

MESTRADO
MULTIMÉDIA - ESPECIALIZAÇÃO EM TECNOLOGIAS

Interfaces Conversacionais – *Chatbot* para a Casa da Música

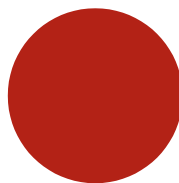
Sara Filipa Gomes Oliveira

M

2017

FACULDADES PARTICIPANTES:

**FACULDADE DE ENGENHARIA
FACULDADE DE BELAS ARTES
FACULDADE DE CIÊNCIAS
FACULDADE DE ECONOMIA
FACULDADE DE LETRAS**



Interfaces Conversacionais – *Chatbot* para a Casa da Música

Sara Filipa Gomes Oliveira

Mestrado em Multimédia da Universidade do Porto

Orientador: Prof.^a Dr.^a Maria Teresa Magalhães da Silva Pinto de Andrade

Junho de 2017

© Sara Filipa Gomes Oliveira, 2017

Interfaces Conversacionais – *Chatbot*
para a Casa da Música

Sara Filipa Gomes Oliveira

Mestrado em Multimédia da Universidade do Porto

Aprovado em provas públicas pelo Júri:

Presidente: Prof. Dr. André Monteiro de Oliveira Restivo

Vogal Externo: Prof. Dr. António Joaquim da Silva Teixeira

Orientador: Prof.^a Dr.^a Maria Teresa Magalhães da Silva Pinto de Andrade

Resumo

Interfaces conversacionais permitem o diálogo entre seres humanos e máquinas, sendo que estas máquinas têm como objetivo tentar substituir os humanos numa componente de transação informacional, a nível de diálogo. Estas interfaces podem ser assistentes de voz, ou *chatbots*. Na primeira componente, o diálogo é realizado através de voz, enquanto que na segunda é através de texto.

Neste contexto, a Xarevision S.A., empresa que desenvolve sistemas de informação para retalho, lançou o projeto de implementação de um protótipo de um *chatbot* a ser utilizado via Facebook Messenger. Simultaneamente, surgiu a oportunidade de colaboração com a Casa da Música, uma instituição relacionada com eventos culturais. A sua agenda cultural tornou-se então a área de aplicação associada a este protótipo.

Com a finalidade de encontrar rapidamente uma solução eficiente, capaz de proporcionar uma boa qualidade de experiência ao utilizador, oferecendo-lhe uma interface simples e intuitiva de utilizar, foi definido que um dos objetivos centrais desta dissertação, seria o de efetuar uma avaliação de interfaces já existentes, quer do ponto de vista tecnológico quer funcional. Identificaram-se quais as vantagens da sua utilização, e a evolução que tem sofrido desde a sua introdução. Também foram identificadas as tecnologias mais utilizadas atualmente para desenvolver este tipo de sistemas. Para além disso, foi desenvolvido um protótipo de um *chatbot* com o particular foco de permitir conhecer a agenda de eventos da Casa da Música e adquirir bilhetes através de uma interface conversacional, de modo a obter resultados experimentais e extrair conclusões.

Foram realizados testes de usabilidade com utilizadores, quanto ao uso do protótipo desenvolvido e os resultados foram bastante positivos. Estes permitiram extrair um nível de aceitação também bastante elevado.

Palavras Chave (Tema): Interfaces conversacionais, *chatbot*, eventos culturais

Palavras Chave (Tecnologias): Facebook Messenger, Api.ai, *web services*

Abstract

Conversational interfaces allow establishing a dialog among humans and machines that endeavor to replace humans as an informational intermediate on a dialog level. These interfaces can be voice assistants or chatbots. When using voice assistants the dialog is established via voice, while chatbots depend on textual methods.

Xarevision S.A., a company related to informational systems for retail, created a project that comprises the development of a chatbot prototype to be used on Facebook Messenger. This inspired the prospect of cooperating with "Casa da Música", an institution related to cultural events. It's the "Casa da Música" cultural agenda that became the application area of this prototype.

In order to identify how to obtain answers in a more quick and effective way while maintaining a user friendly experience, the aim of this dissertation was to characterize conversational interfaces in technological and functional terms. The advantages associated with its use and the evolution ever since its introduction were identified as well as the top technologies used currently to develop this type of systems. This dissertation also had the goal of developing a prototype with the focus of searching and finding events in the agenda of "Casa da Música" and purchased tickets through a conversational interface, with the aim of obtaining experimental results and extract conclusions.

Usability tests were performed with users regarding the use of the developed prototype and the results were very positive. These allowed to extract a level of acceptance also quite high.

Keywords (Theme): Conversational interfaces, chatbot, cultural events

Keywords (Technologies): Facebook Messenger, Api.ai, web services

Agradecimentos

Agradeço a todas as pessoas que possibilitaram e apoiaram a realização desta dissertação.

Ao Eng.º Sílvio Macedo e ao Norberto Amaral, tanto pela orientação como pelo tempo despendido em todas as reuniões ao longo deste percurso. À Ana de Sousa que sempre me ajudou, e foi uma pessoa muito importante neste caminho. A todas as pessoas da Xarevision S.A. que contribuíram para um melhor desenrolar desta dissertação e que intervieram. A todos, um agradecimento também pela ajuda na resolução de problemas e dúvidas surgidas durante todo o processo.

Ao André Alves, ao Nuno Guedes e a todos os envolventes da Casa da Música, pela disponibilidade e interesse que sempre demonstraram, mas também pela ajuda fornecida que foi uma mais-valia.

À orientadora da FEUP, Prof.^a Dr.^a Teresa Andrade, pela orientação, acompanhamento e pela ajuda.

À minha família, namorado e aos meus amigos por toda a confiança que depositaram em mim, todo o apoio e segurança que me deram para continuar e por toda a ajuda durante o meu percurso académico.

Índice

| | |
|--|-----------|
| 1.Introdução | 1 |
| 1.1.Apresentação da Dissertação | 1 |
| 1.1.1.Área de Aplicação | 2 |
| 1.1.2.Apresentação das Organizações Envolvidas | 3 |
| 1.1.3.Colaboradores | 4 |
| 1.1.4.Planeamento da Dissertação..... | 4 |
| 1.2.Objetivos..... | 6 |
| 1.3.Enquadramento | 7 |
| 1.4.Tecnologias Utilizadas | 8 |
| 1.5.Metodologia Utilizada | 9 |
| 1.6.Estrutura do Documento..... | 10 |
| 2.Estado da Arte..... | 13 |
| 2.1.Interfaces Conversacionais | 14 |
| 2.1.1.Evolução das Interfaces Conversacionais | 14 |
| 2.1.2.Vantagens das Interfaces Conversacionais | 16 |
| 2.1.3.Assistentes de Voz..... | 18 |
| 2.1.4.Chatbots | 19 |
| 2.1.5.Síntese do Subcapítulo..... | 19 |
| 2.2.Trabalho Relacionado..... | 21 |
| 2.2.1.1-800 Flowers | 22 |
| 2.2.2.CNN..... | 23 |
| 2.2.3.Hi Poncho | 25 |
| 2.2.4.KLM | 27 |

| | |
|---|-----------|
| 2.2.5.Síntese do Subcapítulo | 29 |
| 2.3.Ferramentas para Criação de Sistemas de Diálogo | 29 |
| 2.3.1.Wit.ai | 29 |
| 2.3.2.Api.ai | 32 |
| 2.3.3.Chatfuel | 36 |
| 2.3.4.Síntese do Subcapítulo..... | 39 |
| 3.Contexto e Descrição Técnica | 41 |
| 3.1.Enquadramento e Detalhes da Dissertação | 41 |
| 3.2.Análise e Modelação | 42 |
| 3.2.1.Análise do Público-alvo | 42 |
| 3.2.2.Requisitos | 50 |
| 3.2.3.Casos de Uso..... | 51 |
| 3.2.4.Diagrama de Atividades | 58 |
| 3.2.5.Diagrama de Arquitetura | 59 |
| 3.3.Síntese do Capítulo | 60 |
| 4.Implementação | 61 |
| 4.1.Api.ai..... | 61 |
| 4.1.1.Criação do Agente | 61 |
| 4.1.2.Criação de Entidades | 63 |
| 4.1.3.Integração com o Facebook Messenger..... | 64 |
| 4.1.4.Criação de <i>Intents</i> | 66 |
| 4.1.5.Ligação com o Servidor..... | 66 |
| 4.2.Back-end..... | 67 |
| 4.2.1.Criação de Back-end em PHP | 67 |
| 4.2.2.Ligação com o Heroku | 68 |
| 4.2.3.Repositório Bitbucket..... | 68 |
| 4.2.4.Receção do Pedido e Envio de Respostas..... | 69 |
| 4.3.Compra de Bilhetes | 70 |

| | |
|---|------------|
| 4.3.1.Entidades..... | 70 |
| 4.3.2.Processo de Compra de Bilhetes | 71 |
| 4.3.3.Formas de Pagamento..... | 74 |
| 4.4.Obter Informações sobre Eventos | 77 |
| 4.4.1.Entidades..... | 77 |
| 4.4.2.Tipos de Pedido..... | 77 |
| 4.4.3.Respostas | 82 |
| 4.5.Obter Informações sobre Artistas..... | 85 |
| 4.5.1.Entidades | 85 |
| 4.5.2.Tipos de Resposta..... | 86 |
| 4.6.Obter Informações sobre a Casa da Música..... | 88 |
| 4.6.1.Entidades..... | 89 |
| 4.6.2.Tipos de Pedido..... | 89 |
| 4.7.Sugestão de Funcionalidades | 96 |
| 4.8.Síntese do Capítulo..... | 97 |
| 5.Testes | 99 |
| 5.1.Testes Operacionais..... | 99 |
| 5.2.Testes Funcionais | 101 |
| 5.3.Testes de Usabilidade | 108 |
| 5.4.Síntese do Capítulo..... | 116 |
| 6.Conclusões | 117 |
| 6.1.Objetivos Realizados | 117 |
| 6.2.Trabalho Futuro..... | 118 |
| 6.3.Apreciação Final | 119 |
| 7.Referências | 121 |
| 8.Anexos | 123 |
| 8.1.Anexo 1 – Tabelas de Frequências (Questionário Inicial) | 123 |
| 8.2.Anexo 2 – Domínios Suportados por cada Idioma..... | 126 |

| | |
|---|-----|
| 8.3.Anexo 3 – Estatísticas Descritivas (Questionário Final) | 128 |
| 8.4.Anexo 4 – Tabelas de Frequências (Questionário Final)..... | 129 |

Lista de Figuras

| | |
|--|----|
| Figura 1 - Planejamento da dissertação (Diagrama de <i>Gantt</i>) | 6 |
| Figura 2 – Interfaces conversacionais com mais destaque desde 1966..... | 15 |
| Figura 3 – Exemplo de <i>chatbot</i> (Microsoft, 2017)..... | 17 |
| Figura 4 – Amazon Echo..... | 18 |
| Figura 5 – Conversa iniciada com o 1-800 Flowers (Facebook Messenger)..... | 22 |
| Figura 6 – Ação de compra de flores (1-800 Flowers)..... | 23 |
| Figura 7 - Conversa iniciada com a CNN (Facebook Messenger)..... | 24 |
| Figura 8 - Ação de seleção da opção: Algo mais (CNN) | 25 |
| Figura 9 - Conversa iniciada com o Hi Poncho (Facebook Messenger) | 26 |
| Figura 10 - Ação de verificação de localização (Hi Poncho) | 27 |
| Figura 11 – Confirmação de reserva e notificações de <i>check-in</i> (KLM)..... | 28 |
| Figura 12 – Receção de atualizações do estado do voo (KLM)..... | 28 |
| Figura 13 – Página pessoal no Wit.ai (aplicações já existentes)..... | 30 |
| Figura 14 – Página associada a um <i>chatbot</i> - Histórias (Wit.ai)..... | 31 |
| Figura 15 – Criação de uma nova história (Wit.ai)..... | 31 |
| Figura 16 – Página do Wit.ai relativa ao conhecimento de entidades..... | 32 |
| Figura 17 – Página inicial da Api.ai após autenticação..... | 33 |
| Figura 18 – Página dos agentes criados (Api.ai)..... | 33 |
| Figura 19 – Página de um <i>intent</i> criado (Api.ai)..... | 34 |
| Figura 20 – Página de uma entidade criada (Api.ai) | 35 |
| Figura 21 – Página dos domínios do <i>chatbot</i> (Api.ai)..... | 35 |
| Figura 22 – Página das integrações associadas ao <i>chatbot</i> (Api.ai)..... | 36 |
| Figura 23 – Dashboard do Chatfuel | 37 |
| Figura 24 – Página de definição da conversa associada a um robot (Chatfuel)..... | 37 |

| | |
|--|----|
| Figura 25 – Definição de uma mensagem (Chatfuel) | 38 |
| Figura 26 – Conversa criada no Chatfuel vistas no Facebook Messenger | 38 |
| Figura 27 – <i>Plugins</i> associados ao Chatfuel | 39 |
| Figura 28 – Percentagem de compra de bilhetes (dados estatísticos) | 44 |
| Figura 29 - Média de respostas (Q01 – Q06) - questionário inicial | 46 |
| Figura 30 – Percentagem de respostas à questão Q07 - questionário inicial | 48 |
| Figura 31 – Percentagem de respostas à questão Q08 - questionário inicial | 48 |
| Figura 32 - Percentagem de respostas à questão Q09 - questionário inicial | 49 |
| Figura 33 – Diagrama de casos de uso | 52 |
| Figura 34 – Diagrama de atividades do fluxo de interações do utilizador/agente | 58 |
| Figura 35 – Fluxo de interações do utilizador com o <i>chatbot</i> | 59 |
| Figura 36 – Diagrama de arquitetura | 59 |
| Figura 37 – Página de definições do agente criado | 63 |
| Figura 38 – Entidade <i>@artist</i> | 64 |
| Figura 39 – <i>Webhooks</i> associados à aplicação CM_Sara | 64 |
| Figura 40 – Integração com o Facebook Messenger (Api.ai) | 65 |
| Figura 41 – Acesso ao <i>token</i> da página de Facebook | 65 |
| Figura 42 – Ativação do <i>webhook</i> | 66 |
| Figura 43 – Ativação do uso do <i>webhook</i> para um <i>intent</i> | 67 |
| Figura 44 – Comandos para envio de alterações para o Heroku | 68 |
| Figura 45 – <i>Commits</i> no repositório do Bitbucket | 68 |
| Figura 46 – Verificação do <i>intent</i> invocado | 69 |
| Figura 47 – Definição das respostas (<i>back-end</i>) | 70 |
| Figura 48 - <i>Intent buyTicketsWithContext</i> | 71 |
| Figura 49 – Pedido de bilhetes com contexto | 72 |
| Figura 50 - <i>Intent buyTicketsWithoutContext</i> | 72 |
| Figura 51 – Compra de bilhetes sem contexto | 73 |
| Figura 52 - Compra de bilhetes para um evento específico | 73 |

| | |
|--|----|
| Figura 53 – Mensagem de escolha da forma de pagamento | 74 |
| Figura 54 – <i>Intent buyTicketsCardInsertData</i> | 75 |
| Figura 55 – Compra de bilhetes (cartão de crédito/débito) | 75 |
| Figura 56 - <i>Intent buyTicketsATM</i> | 76 |
| Figura 57 - Compra de bilhetes (ATM)..... | 76 |
| Figura 58 – <i>Intent getInfoEventWithoutContext</i> | 78 |
| Figura 59 – <i>Intent getInfoEventWithContext</i> | 78 |
| Figura 60 – Pergunta definida para o utilizador dar o nome do evento | 79 |
| Figura 61 – <i>Intent getDateEventWithoutContext</i> | 80 |
| Figura 62 – <i>Intent getDateEventWithContext</i> | 80 |
| Figura 63 – <i>Intent getRoomEventWithContext</i> | 81 |
| Figura 64 – <i>Intent getPriceEventWithoutContext</i> | 82 |
| Figura 65 – Método de devolução do preço de um evento | 82 |
| Figura 66 – Informações sobre um evento não encontrado | 83 |
| Figura 67 – Pedido de informações e data de um evento (I) | 83 |
| Figura 68 - Pedido de informações e data de um evento (II)..... | 84 |
| Figura 69 - Pedidos de data de um evento (sem contexto) | 84 |
| Figura 70 - Pedidos de informação sobre a sala e preço de um evento | 85 |
| Figura 71 – Informação sobre artistas (tipo de resposta I) | 86 |
| Figura 72 – Mensagem cortada em várias respostas | 86 |
| Figura 73 - Método que delimita o número de caracteres de uma mensagem | 87 |
| Figura 74 - Informação sobre artistas (tipo de resposta II)..... | 87 |
| Figura 75 - Informação sobre artistas (tipo de resposta III) | 87 |
| Figura 76 - Informação sobre artistas (tipo de resposta IV) | 88 |
| Figura 77 – Entidade <i>CmName</i> | 89 |
| Figura 78 – <i>Intent getInfoCM</i> | 90 |
| Figura 79 – Pedido de detalhes sobre a Casa da Música (I) | 90 |
| Figura 80 - Pedido de detalhes sobre a Casa da Música (II)..... | 91 |

| | |
|---|-----|
| Figura 81 - Pedido de detalhes sobre a Casa da Música (III) | 91 |
| Figura 82 – Respostas a dar ao utilizador após pedido de morada da instituição | 92 |
| Figura 83 – Pedido da morada | 92 |
| Figura 84 – Contextos de <i>getAddressCM</i> | 93 |
| Figura 85 – Contextos de <i>getHelpCM_yes</i> | 93 |
| Figura 86 – Contextos de <i>getHelpCM_no</i> | 93 |
| Figura 87 – Opção de veículo para indicar ao utilizador como chegar à instituição | 94 |
| Figura 88 – Resposta consoante o tipo de veículo escolhido (I) | 94 |
| Figura 89 - Resposta consoante o tipo de veículo escolhido (II) | 95 |
| Figura 90 - Resposta consoante o tipo de veículo escolhido (III) | 95 |
| Figura 91 – Conversa relativa ao horário da Casa da Música | 96 |
| Figura 92 – Sugestão de funcionalidades | 97 |
| Figura 93 – Tempos de resposta (segundos) associados a cada funcionalidade | 100 |
| Figura 94 - Percentagem de respostas à questão Q01 - questionário final | 111 |
| Figura 95 – Percentagem de respostas à questão Q02 - questionário final | 111 |
| Figura 96 - Percentagem de respostas à questão Q03 - questionário final | 112 |
| Figura 97 - Percentagem de respostas à questão Q04 - questionário final | 113 |
| Figura 98 - Percentagem de respostas à questão Q05 - questionário final | 113 |
| Figura 99 - Percentagem de respostas à questão Q06 - questionário final | 114 |

Lista de Tabelas

| | |
|--|----|
| Tabela 1 – Planeamento da dissertação (lista de tarefas) (I)..... | 4 |
| Tabela 2 - Planeamento da dissertação (lista de tarefas) (II)..... | 5 |
| Tabela 3 – Questões de investigação | 7 |
| Tabela 4 – <i>Software</i> utilizado | 8 |
| Tabela 5 – Ferramentas <i>web</i> utilizadas | 9 |
| Tabela 6 – Diferenças entre as interfaces conversacionais (antigas vs atuais) | 16 |
| Tabela 7 – Tabela de comparação entre Amazon Echo e Google Home (I)..... | 20 |
| Tabela 8 – Tabela de comparação entre Amazon Echo e Google Home (II) | 21 |
| Tabela 9 – Comparação entre as ferramentas de criação de <i>chatbots</i> (I) | 39 |
| Tabela 10 – Comparação entre as ferramentas de criação de <i>chatbots</i> (II) | 40 |
| Tabela 11 – Acesso ao sítio <i>web</i> da Casa da Música (dados estatísticos) | 43 |
| Tabela 12 – Questões de componente quantitativa - questionário inicial | 45 |
| Tabela 13 - Tabela descritiva (Q01 – Q06) - questionário inicial..... | 46 |
| Tabela 14 – Questões relativas a frequências - questionário inicial..... | 47 |
| Tabela 15 – Funcionalidades sugeridas (I)..... | 49 |
| Tabela 16 – Funcionalidades sugeridas (II)..... | 50 |
| Tabela 17 – Lista de requisitos (I) | 50 |
| Tabela 18 – Lista de requisitos (II) | 51 |
| Tabela 19 – Descrição do caso de uso 1 | 52 |
| Tabela 20 - Descrição do caso de uso 1.1 | 53 |
| Tabela 21 – Descrição do caso de uso 1.2..... | 53 |
| Tabela 22 – Descrição do caso de uso 1.3..... | 54 |
| Tabela 23 – Descrição do caso de uso 2..... | 54 |

| | |
|---|-----|
| Tabela 24 – Descrição do caso de uso 2A | 54 |
| Tabela 25 – Descrição do caso de uso 2B | 55 |
| Tabela 26 – Descrição do caso de uso 2C | 55 |
| Tabela 27 – Descrição do caso de uso 2D | 55 |
| Tabela 28 – Descrição do caso de uso 3 | 56 |
| Tabela 29 – Descrição do caso de uso 3.1 | 56 |
| Tabela 30 – Descrição do caso de uso 4 | 56 |
| Tabela 31 – Descrição do caso de uso 4A | 57 |
| Tabela 32 – Descrição do caso de uso 4B | 57 |
| Tabela 33 – Descrição do caso de uso 4B.1 | 57 |
| Tabela 34 – Descrição do caso de uso 4C | 58 |
| Tabela 35 – Teste do pedido de informação histórica da Casa da Música | 101 |
| Tabela 36 – Teste relativo ao pedido da morada da Casa da Música | 102 |
| Tabela 37 - Teste relativo à ajuda de indicação de como chegar à Casa da Música | 102 |
| Tabela 38 – Teste à devolução direta de informação de um artista. | 103 |
| Tabela 39 - Teste ao pedido de informação de um artista não existente | 103 |
| Tabela 40 – Teste ao pedido de informação de um artista sem detalhes | 103 |
| Tabela 41 – Teste ao pedido de informação de um artista sem detalhes em inglês | 104 |
| Tabela 42 – Teste relativo à tradução dos detalhes de um artista | 104 |
| Tabela 43 - Teste relativo ao pedido da informação de um evento | 105 |
| Tabela 44 - Teste relativo ao pedido da informação de um evento não encontrado | 105 |
| Tabela 45 - Teste relativo ao pedido da compra de bilhetes | 106 |
| Tabela 46 - Teste relativo à aceitação da compra de bilhetes | 106 |
| Tabela 47 – Teste relativo à escolha de forma de pagamento | 107 |
| Tabela 48 - Teste relativo à inserção dos dados para efetuar compra de bilhetes | 107 |
| Tabela 49 – Teste relativo a um pedido realizado com negação | 108 |
| Tabela 50 – Passos realizados em cada teste de usabilidade | 109 |
| Tabela 51 – Lista de tarefas para os testes de usabilidade | 109 |

| | |
|---|-----|
| Tabela 52 – Questões de componente quantitativa - questionário final..... | 110 |
| Tabela 53 – Questões de escolha múltipla - questionário final..... | 112 |
| Tabela 54 – Questões de resposta aberta - questionário final..... | 114 |
| Tabela 55 – Melhorias sugeridas pelos participantes | 115 |
| Tabela 56 - Particularidades que o utilizador gostou | 115 |
| Tabela 57 – Objetivos e sua concretização..... | 118 |

Abreviaturas e Símbolos

| | |
|------|---|
| AIML | Artificial Intelligence Markup Language |
| API | Application Programming Interface |
| CTR | Click-Through Rate |
| JSON | JavaScript Object Notation |
| MIT | Massachusetts Institute of Technology |
| NLP | Neuro-Linguistic Programming |
| SDK | Software Development Kit |
| UML | Unified Modeling Language |
| URL | Uniform Resource Locator |
| XML | eXtensible Markup Language |

1. Introdução

Neste capítulo pretende-se realizar uma breve introdução e enquadramento da dissertação realizada, começando por ser feita uma apresentação ao projeto, tendo em conta a empresa relacionada com a mesma (Xarevision S.A.). Este subcapítulo integra a área de aplicação associada, apresentação das organizações que colaboraram durante a realização desta dissertação, os colaboradores mais específicos que contribuíram de forma direta para o desenrolar da mesma, assim como o planeamento do trabalho.

De seguida são apresentados os objetivos desta dissertação, assim como realizado um enquadramento da mesma.

Na introdução são também referidas todas as tecnologias e as metodologias utilizadas neste âmbito, e também realizado um breve resumo de toda a estrutura do relatório.

1.1. Apresentação da Dissertação

Este documento tem como objetivo descrever e documentar a dissertação desenvolvida no âmbito da unidade curricular denominada por Dissertação, do ano letivo 2016/2017, do Mestrado em Multimédia (MM), da Faculdade de Engenharia da Universidade do Porto (FEUP). Esta unidade curricular tem como finalidade proporcionar aos alunos a possibilidade de pesquisa e análise de potenciais soluções, desenvolvimento de uma solução e avaliação da mesma, tendo como ponto de partida um problema a resolver.

Esta dissertação foi realizada numa organização externa à FEUP, a Xarevision, S.A, e tem como principal objetivo analisar, com detalhe, ferramentas disponíveis que atualmente permitam criar um sistema de diálogo, tendo sempre como base o tema de interfaces conversacionais.

Interfaces conversacionais são caracterizadas por permitirem o diálogo entre seres humanos e máquinas, sendo que estas tentam reproduzir humanos no contexto da interação. A pessoa passa a interagir quase como se estivesse a comunicar com outra pessoa, ao invés da comunicação entre

2 - Introdução

humanos e computadores, ser realizada diretamente através dos seus domínios não humanos (inserindo comandos específicos, por exemplo).

Assim, depois de alguma análise, decidiu-se envergar pela agenda cultural da Casa da Música, para o desenvolvimento do sistema de diálogo: *chatbot* para a Casa da Música. Isto porque se concluiu que seria um bom alvo de estudo, uma vez que para esta área não há nada semelhante. O intuito final é adquirir bilhetes para os eventos que possuem, através de uma interface conversacional. Tendo sempre a possibilidade de outros fins, como por exemplo, a possibilidade de obter informações artistas, espetáculos e sobre a própria instituição.

Até agora, as pessoas que pretendem comprar bilhetes ou obter informações, necessitam visitar o sítio *web* da organização e procurarem o que pretendem. Com isto acabam por gastar algum do seu tempo, enquanto que, tendo uma alternativa mais facilitada, podiam despende muito menos. Conversando diretamente com alguém, que neste caso seria um agente (um *robot*), seria muito mais fácil obter as informações pretendidas ou mesmo realizar as tarefas ambicionadas.

Com este mesmo propósito, inicialmente será então realizado um levantamento de informação, percecionando quais as ferramentas atuais que permitem a criação de um sistema de diálogo integrado no Facebook Messenger, investigação de trabalho já existente semelhante, desenvolvimento de um protótipo simples com algumas funcionalidades, assim como análise dos resultados obtidos.

É de acrescentar que todo este trabalho é também importante para a empresa para que fiquem a par das mais atuais tecnologias, tendo em conta um tema com tanto impacto atualmente e futuramente, que tem estado em constante evolução e ainda tem tanto por onde evoluir.

1.1.1. Área de Aplicação

Tendo como tema base interfaces conversacionais, foi necessária a escolha de uma área de aplicação em particular, de modo a ser possível ter um problema mais concreto, assim como uma possível área de desenvolvimento.

As interfaces conversacionais têm vindo a aparecer em vários contextos, e havia vários por onde selecionar. Nas reuniões conjuntamente com a empresa falou-se da possibilidade de uma interface para gestão de salas de reuniões, ou mesmo um sistema de diálogo construído para culinária, com o objetivo de ajudar o cozinheiro a preparar refeições consoante uma lista de ingredientes específicos. Contudo, optou-se pela agenda cultural da Casa da Música, com o objetivo final de adquirir bilhetes para os espetáculos existentes na instituição. Para além de ser um desafio ambicioso, é algo que ainda não existe e que permite explorar vários casos que podem ser úteis para as pessoas que acabam por estar ligadas à Casa da Música, ou seja, o seu público, os seus espectadores.

1.1.2. Apresentação das Organizações Envolvidas

Esta dissertação tem como instituição de acolhimento a empresa Xarevision, S.A., mas como a área de aplicação escolhida está relacionada com a Casa da Música, também esta passou a ser uma organização envolvida na dissertação.

Assim sendo, será feita uma apresentação a cada uma das instituições.

1.1.2.1. Xarevision S.A.

Xarevision (Xarevision, S.A., 2015) é uma empresa líder em tecnologias para retalho.

Como especialista em áreas tais como: televisão corporativa, sinalização digital e gestão de filas, a Xarevision administra redes digitais de dispositivos geridos centralmente. Neste contexto, há vários anos, esta empresa opera a maior rede digital em loja, em Portugal, atingindo mais de 40% da população ativa no país.

Criaram também meios de comunicação interativos altamente diferenciados, que incluem recursos, tais como reconhecimento de gestos, georreferenciação e endereçamento individual.

Estimulados pelos requisitos crescentes e desafiadores do maior retalhista português, expandiram a atividade para outros domínios emergentes (*wearables*¹ para gestão de tarefas e suporte de vendas; solução de rastreamento de clientes em lojas e análise de inteligência artificial).

Dada a grande importância do assunto, foi criado um grupo de trabalho dentro da Xarevision, mas apenas dedicados a *Big Data*², com o objetivo de enfrentar desafios, usando novos paradigmas. Como resultado, a equipa tem vindo a crescer, e agora abrange as áreas de infraestrutura, arquitetura, programação, operações, análise e *machine learning*³.

1.1.2.2. Casa da Música

Imaginada para assinalar o ano festivo de 2001, em que a cidade do Porto foi Capital Europeia da Cultura, a Casa da Música (Casa da Música, 2016) é o primeiro edifício construído em Portugal exclusivamente dedicado à Música, seja no domínio da apresentação e fruição pública, seja no campo da formação artística e da criação.

¹ Dispositivos tecnológicos que podem ser utilizados como peças de vestuário.

² Refere-se a um grande conjunto de dados.

³ Subcampo de ciências de computadores que evoluiu do estudo de reconhecimento de padrões e da teoria da aprendizagem computacional em inteligência artificial.

4 - Introdução

O projeto Casa da Música foi definido em 1999, como resultado de um concurso internacional de arquitetura que escolheu a solução apresentada por Rem Koolhaas - *Office for Metropolitan Architecture*. As escavações iniciaram-se ainda em 1999, no espaço da antiga Remise do Porto na Rotunda da Boavista, e a Casa da Música foi inaugurada na Primavera de 2005, no dia 15 de Abril.

1.1.3. Colaboradores

No âmbito desta dissertação, estiveram envolvidos colaboradores tanto da parte da empresa Xarevision, S.A., como da própria Casa da Música.

Para além do CEO da primeira empresa, Eng.º Sílvio Macedo, também outras duas pessoas estiveram incluídas no desenrolar desta dissertação: Norberto Amaral (Cofundador e business developer) e Maria João (designer). Também outros elementos da empresa contribuíram ao longo deste percurso, ajudando sempre que necessário.

Já da Casa da Música, os colaboradores foram o responsável pelo marketing digital, André Alves, o responsável de bilheteira e atendimento ao cliente, José Ribeiro, e da área de sistemas de informação, Nuno Guedes.

1.1.4. Planeamento da Dissertação

À dissertação foi delineado um planeamento inicial, para servir como guia do trabalho necessário a desenvolver, com datas estipuladas a seguir.

Foi então dividido em várias fases, tendo associadas datas de início e término das mesmas. Para melhor perceção, na Tabela 1 e Tabela 2, é apresentada essa divisão de fases e tarefas, com data de início e data de fim, consoante o planeamento definido, assim como um diagrama de *Gantt* (Figura 1).

Tabela 1 – Planeamento da dissertação (lista de tarefas) (I)

| Nome | Data de Início | Data de Fim |
|--|----------------|-------------|
| Escrita da proposta de dissertação | 19-09-2016 | 17-10-2016 |
| Estado da arte | 19-09-2016 | 30-01-2017 |
| Escrita do relatório | 25-10-2016 | 14-06-2017 |
| Apresentação e entrega do estado da arte | 30-01-2017 | 03-02-2017 |
| Análise e modelação | 06-02-2017 | 24-02-2017 |
| Implementação do protótipo | 27-02-2017 | 28-04-2017 |
| Testes de desenvolvimento | 27-02-2017 | 28-04-2017 |

Tabela 2 - Planeamento da dissertação (lista de tarefas) (II)

| Nome | Data de Início | Data de Fim |
|--|----------------|-------------|
| Apresentação intermédia | 03-04-2017 | 09-04-2017 |
| Testes operacionais, funcionais e de usabilidade | 01-05-2017 | 26-05-2017 |
| Validação pelos orientadores | 15-06-2017 | 30-06-2017 |
| Defesas | 12-07-2017 | 19-07-2017 |
| Entrega da documentação final | 12-07-2017 | 28-07-2017 |

Desde 19 de setembro até 17 de outubro de 2016, será escrita a proposta de dissertação para entrega através do Sigarra, de modo a ser aprovada. A partir do mesmo dia de início que a tarefa anterior, até 30 de janeiro de 2017, será feita uma pesquisa bibliográfica, assim como escrita no relatório sobre capítulo referente ao estado da arte e apresentação.

Desde 25 de outubro de 2016 até 14 de junho de 2017, será a época de escrita do relatório final.

Na semana de 30 de janeiro a 3 de fevereiro, será realizada uma apresentação inicial para o coordenador de especialização ou um seu representante, e um segundo elemento selecionado pelo coordenador, assim como entrega desta mesma secção.

Nas semanas de 6 a 24 de Fevereiro será a fase de análise e modelação, tendo em conta o levantamento de requisitos finais, casos de uso e diagramas necessários.

De 27 de fevereiro a 28 de abril, a implementação do protótipo será desenvolvida. Também neste período de tempo serão efetuados testes a nível de desenvolvimento.

Na semana de 3 a 9 de abril, será realizada uma apresentação intermédia da dissertação, novamente para o coordenador de especialização e um segundo elemento selecionado pelo mesmo.

Seguidamente, de 1 a 26 de maio, outros testes serão concretizados: testes operacionais, funcionais e de usabilidade.

A semana de 15 a 30 de junho, prende-se uma validação da dissertação, por parte dos orientadores.

Na semana de 12 a 19 de julho, encontrar-se-ão as apresentações finais, ou seja, as defesas.

Por fim, de 12 a 28 de julho, será entregue toda a documentação final.

6 - Introdução

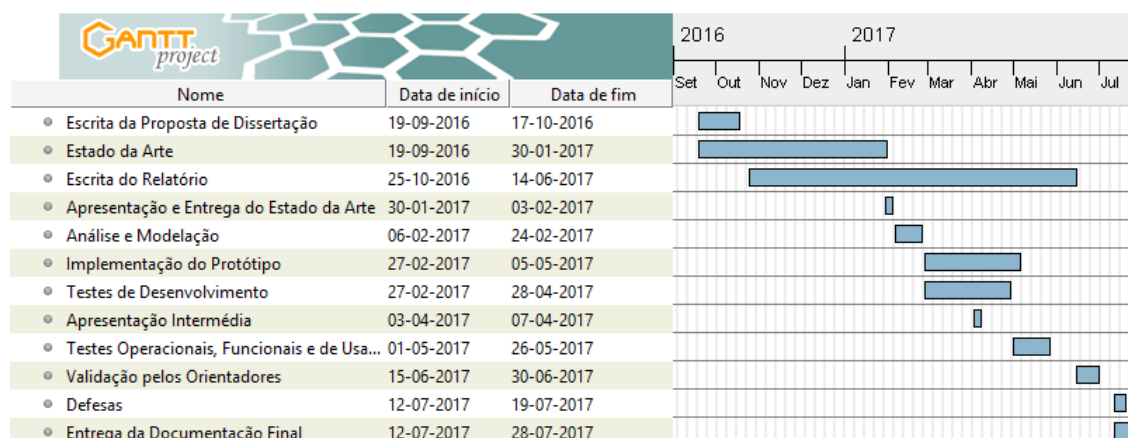


Figura 1 - Planeamento da dissertação (Diagrama de Gantt)

1.2. Objetivos

O objetivo desta dissertação é efetuar uma caracterização detalhada de interfaces conversacionais, em termos tecnológicos e funcionais. Pretende-se identificar quais as vantagens da sua utilização e a evolução que tem sofrido desde a sua introdução. Deverão ser identificadas as tecnologias mais utilizadas atualmente para desenvolver este tipo de sistemas, capacidades que oferecem, facilidade e âmbito de utilização, desempenho oferecido, entre outros. Espera-se conseguir efetuar quer uma análise exaustiva, quer pequenos desenvolvimentos que possibilitem a sua experimentação, de forma a obter resultados conclusivos.

Essa experimentação terá também como objetivo desenvolver um protótipo que permita mostrar a possibilidade de compra de bilhetes para espetáculos na Casa da Música, integrando uma interface conversacional e recorrendo apenas a uma dessas ferramentas exploradas.

Após este desenvolvimento, haverá uma fase de análise dos resultados obtidos, consoante o que se desenvolveu.

Assim sendo, as questões de investigação associadas a esta dissertação, são apresentadas através da Tabela 3.

Tabela 3 – Questões de investigação

| Questão |
|---|
| 1. Como é que se processou a evolução das interfaces conversacionais? |
| 2. Como é possível as interfaces conversacionais terem utilidade para as pessoas no contexto das suas interações com produtos e serviços? |
| 3. Quais as ferramentas que melhor permitem a criação de um sistema conversacional? |
| 4. Qual o nível de aceitação pelos utilizadores de um sistema conversacional baseado numa área de aplicação específica, relativamente às suas necessidades? |

A primeira questão será explorada no levantamento do estado da arte (capítulo 2), pois um dos pontos abordados, passa pela evolução das interfaces conversacionais (subcapítulo 2.1.1). Assim sendo, a questão fica enquadrada no subcapítulo referido.

A segunda questão também será abordada no estado da arte, nas vantagens das interfaces conversacionais, assim como em casos mais específicos explorados na investigação de trabalho relacionado (subcapítulo 2.2). Também na definição da área de aplicação (subcapítulo 1.1.1) foi identificada uma possível interação entre o utilizador e um serviço. Com o desenvolvimento do protótipo será mostrada a utilidade desse propósito.

A terceira questão mais uma vez será abordada no estado da arte, pois será feita uma investigação quanto às ferramentas mais utilizadas para desenvolvimento de *chatbots* (subcapítulo 2.3). Com o desenvolvimento do protótipo será demonstrada uma das ferramentas, após análise de cada uma no capítulo do estado da arte.

Já a quarta questão vai ser explorada através de um questionário inicial aos possíveis utilizadores (subcapítulo 3.2.1.2), assim como através dos testes a serem realizados no final, mais concretamente, através dos testes de usabilidade (subcapítulo 5.3).

1.3. Enquadramento

Como ponto inicial, esta dissertação parte de um tema que está muito em voga e que tem tido muito destaque atualmente. O tema de interfaces conversacionais tem vindo a evoluir cada vez mais rápido, e ainda tem muito por onde progredir. Este tema está longe da estagnação, uma vez que ainda existe um longo percurso de pesquisa e de desenvolvimento.

De modo a entender melhor esta área e a se conseguir uma pesquisa mais aprofundada e com detalhe, foi definida uma área de aplicação: a agenda cultural da Casa da Música. Com o intuito de perceber o nível de aceitação dos utilizadores quanto a uma interface conversacional, o objetivo é criar um novo cenário de compra de bilhetes para eventos na instituição e eventuais pedidos de informação: desenvolvimento de um agente através de um *chatbot* no Facebook Messenger.

8 - Introdução

Uma vez que as pessoas que necessitam de informação, por parte da Casa da Música, vão interagir com um agente criado, não há necessidade de ter mais uma pessoa a despende parte do seu tempo a realizar tarefas de informação por *chat*.

Este trabalho também age na simplificação das vidas das pessoas, obtendo respostas mais rápidas e eficazes consoante aquilo que pretendem.

Portanto, os interessados conseguem obter todas as informações que pretendem relativamente à Casa da Música, assim como à agenda cultural e espetáculos da mesma, por exemplo. É uma alternativa bastante mais simples, do ponto de vista do utilizador. Em vez de procurarem toda esta informação através do sítio *web* da instituição, apenas precisam de conversar com alguém (agente) e questionar.

1.4. Tecnologias Utilizadas

Várias tecnologias foram utilizadas ao longo do desenvolvimento desta dissertação. Assim sendo, as tabelas seguintes (Tabela 4 e Tabela 5) fazem uma divisão dessas tecnologias por categorias.

A Tabela 4 encontra-se descrito o *software* utilizado para elaboração da dissertação.

Tabela 4 – *Software* utilizado

| <i>Software</i> | Descrição | Versão |
|---------------------|---|----------|
| Dropbox | Serviço de armazenamento e partilha de arquivos. | 22.4.24 |
| IBM SPSS Statistics | Programa que oferece capacidades estatísticas e analíticas avançadas, de modo a se conseguir compreender de forma profunda e precisa, os dados e proporcionar uma melhor toma de decisão. | 22.0.0.0 |
| Gantt Project | Programa de gestão que permite realizar diagramas de Gantt para projetos de calendarização de tarefas e recursos. | 2.8.1 |
| PHP Storm IDE | Ambiente de desenvolvimento utilizado para a linguagem PHP, HTML e Javascript. | 2017.1 |
| Tortoise Git | Ferramenta de ambiente gráfico que permite o controlo de versões de um projeto. | 1.8.7.0 |

Na Tabela 5 são indicadas quais as ferramentas *web* que foram utilizadas no âmbito do desenvolvimento desta dissertação.

Tabela 5 – Ferramentas *web* utilizadas

| Ferramenta | Descrição |
|--------------------|--|
| Api.ai | Ferramenta que permite a criação de interfaces conversacionais. |
| Bitbucket | Sistema de controlo de versões de um projeto. |
| Draw.io | Ferramenta que permite realizar diagramas <i>online</i> . |
| Facebook Messenger | Ferramenta oficial do Facebook, que permite criar conversas de texto com todos os amigos existentes na rede social. |
| Google Translate | Serviço gratuito da Google que traduz instantaneamente palavras, frases e até páginas <i>web</i> entre português e mais de 100 outros idiomas. |
| Heroku | Plataforma <i>cloud</i> que suporta várias linguagens de programação e é usada como um modelo de desenvolvimento de aplicações <i>web</i> . |
| PlantText | Ferramenta que permite realizar diagramas UML <i>online</i> , através de texto. |

1.5. Metodologia Utilizada

Durante o decorrer desta dissertação, será adotada uma metodologia dividida em várias fases.

Numa primeira fase, será realizado o levantamento do estado da arte, de forma a entender em que consistem as interfaces conversacionais, qual o objetivo das mesmas, assim como as vantagens associadas ao uso destas e como evoluíram desde o seu surgimento. Ainda dentro desta fase, será realizada uma investigação relacionada com interfaces conversacionais semelhantes ao protótipo que se pretende desenvolver e uma seleção das ferramentas que permitem a criação de *chatbots*, e que atualmente são as mais utilizadas.

Na seguinte etapa será feita uma escolha tecnológica, consoante aquilo que já foi definido em subcapítulos anteriores (1.1 e 1.2). Para uma maior definição e explanação técnica, será também tida em conta uma fase de análise e modelação. Aqui, primeiramente será realizada uma análise ao público-alvo; que envolverá um questionário inicial, definição dos requisitos do sistema, de forma a descrever todas as tarefas necessárias; definição dos casos de uso que refletirão as funções pretendidas, associadas ao utilizador; desenho de um diagrama de atividades, de forma a estabelecer o fluxo de interações entre o utilizador e o agente; e por fim, o desenho de um diagrama de atividades para definir a estrutura do sistema.

A terceira etapa consistirá na implementação do protótipo, e em simultâneo na realização de testes de desenvolvimento para perceber se tudo funciona como o esperado.

Numa última fase serão realizados testes finais de modo a entender qual a prestação da aplicação (testes operacionais), se consegue oferecer todas as funcionalidades esperadas (testes funcionais), e testes com os utilizadores para perceber a sua opinião relativamente a esta área de trabalho (testes de usabilidade).

Durante todas estas etapas, serão definidas semanalmente todas as tarefas a desempenhar, de modo a existir uma maior divisão das mesmas, assim como um maior controlo do tempo.

Concluindo, a metodologia baseia-se no seguimento das seguintes fases: análise, *design*, implementação e testes de desenvolvimento e testes finais (operacionais, funcionais e de usabilidade). Todas estas fases permitem obter conclusões quanto aos objetivos iniciais, qual o trabalho necessário futuro e uma apreciação de um ponto de vista mais pessoal.

1.6. Estrutura do Documento

Este documento apresenta-se com uma estrutura dividida em seis capítulos principais: Introdução, Estado da Arte, Contexto e Descrição Técnica, Implementação, Testes e Conclusões.

Inicia-se com o capítulo da Introdução, no qual é apresentada a dissertação. Essa apresentação inclui a identificação da área de aplicação selecionada, uma apresentação das organizações que de algum modo estão relacionadas a esta dissertação, identificação dos colaboradores associados, assim como apresentação do planeamento do trabalho. Das organizações apresentadas, apenas uma é a instituição de acolhimento. Para além disso também são descritos neste capítulo, os objetivos da dissertação, é realizado um enquadramento da mesma, assim como as tecnologias e a metodologia utilizadas.

No capítulo do Estado da Arte são apresentados os resultados da investigação sobre o tema de interfaces conversacionais, tendo em conta a sua evolução e importância, trabalho relacionado e ferramentas para criação de sistemas de diálogo. No final de cada subcapítulo é apresentada uma síntese.

Já no capítulo do Contexto e Descrição Técnica, é feito um enquadramento mais pormenorizado desta dissertação e composta a secção de análise e modelação, antes de se partir para a implementação do protótipo a ser desenvolvido. Esta secção contém uma análise quanto ao público-alvo, o levantamento de requisitos, descrição dos casos de uso presentes num diagrama definido e apresentados outros dois diagramas: um diagrama de atividades do fluxo geral do protótipo a desenvolver e um diagrama da arquitetura do sistema do protótipo funcional que será desenvolvido.

No capítulo seguinte, referente à Implementação, é apresentada a fase de implementação do protótipo definido no âmbito desta dissertação, relacionado com um *chatbot* para a Casa da Música. São também identificados os pontos mais importantes a ter em conta neste processo de desenvolvimento do mesmo.

No capítulo dos Testes são descritos todos os tipos de testes que foram efetuados após a implementação do protótipo: testes operacionais, testes funcionais e testes de usabilidade.

Estes três últimos capítulos apresentam sínteses finais, para uma melhor compreensão do que foi referido.

Por fim, são apresentadas as Conclusões retiradas após término das fases de pesquisa, desenvolvimento e testes. Encontram-se descritos os objetivos realizados no âmbito desta dissertação, caracterizando o que foi ou não executado, indicação de trabalho futuro, e por fim, uma apreciação final.

Este documento contém também uma lista bibliográfica (Referências) e anexos no final do documento (Anexos).

2. Estado da Arte

Durante a fase inicial da dissertação, foi realizada uma pesquisa quanto ao tema de interfaces conversacionais, para assim se perceber se realmente era algo importante para as empresas e os seus clientes, por exemplo, e no que é que realmente consistia, em termos tecnológicos e funcionais. Deste modo concluiu-se que uma das vantagens do uso de interfaces conversacionais é a comunicação entre humanos e máquinas ser realizada através de linguagem natural, tornando-se numa tarefa muito mais fácil para o utilizador em questão. A pesquisa consistiu também na análise de trabalhos relacionados.

Passada esta fase de pesquisa e tendo em conta que existem formas de criar agentes para interfaces conversacionais, passou-se então a uma pesquisa mais focada nas ferramentas existentes no mercado para essa criação de sistemas de diálogo. Este estudo permitiu assim identificar as tecnologias mais utilizadas atualmente, capacidades que oferecem, facilidade e âmbito de utilização, desempenho oferecido, entre outros.

Após analisadas as ferramentas mais adequadas no âmbito desta dissertação (Wit.ai, Api.ai e Chatfuel), foi selecionada apenas uma delas para desenvolvimento de um protótipo de exemplificação, consoante a área de aplicação elegida. Essa área é a agenda cultural da Casa da Música, e permite assim ter o objetivo de possibilitar ao utilizador conhecer a agenda de eventos e adquirir bilhetes através de uma interface conversacional. Essa seleção será devidamente esclarecida no capítulo 2.3.4.

Neste capítulo está então integrado o tema de interfaces conversacionais, tendo em conta a sua evolução, importância e qual a sua ligação com assistentes de voz, assim como *chatbots*. São também exploradas aplicações existentes e semelhantes ao que se pretende desenvolver, ou seja, exposição de trabalhos relacionados. E por fim, apresentadas algumas ferramentas para criação de um sistema de diálogo, já referidas anteriormente.

2.1. Interfaces Conversacionais

Interface conversacional é uma interface que permite o diálogo entre seres humanos e máquinas, com estas a tentar imitar humanos no contexto da interação. Em vez da comunicação entre humanos e computadores ser realizada diretamente através dos seus domínios não humanos (inserindo comandos específicos, por exemplo), a pessoa interage quase como se estivesse a comunicar com outra pessoa. As máquinas tentam substituir os humanos, numa componente de transação informacional, a nível de diálogo.

Esta é uma área muito atual e em constante mudança, que tem vindo a evoluir cada vez mais. (subcapítulo 2.1.1).

Existem dois tipos de interfaces conversacionais: assistentes de voz, onde a conversa é realizada por voz, ou *chatbots*, onde a conversa é escrita (Brownlee, 2016). Ambos os tipos serão abordados nos subcapítulos 2.1.3 e 2.1.4.

No final será feita uma síntese de todas as abordagens realizadas.

2.1.1. Evolução das Interfaces Conversacionais

Em 1968, Winograd, um cientista da computação, professor da Universidade Stanford, e codiretor do grupo de interação humano-computador de Stanford, decidiu criar um programa com o objetivo de conversar, via *prompt*⁴, utilizando linguagem natural. Otimista depois da criação do primeiro *chatbot* (ELIZA (Wallace, 1999-2006) em 1966) por Joseph Weizenbaum (professor no MIT), programou o seu *chatbot* (SHRDLU (Blom & Thorsen, 2013)) com um *cartoon* que repetia parte das questões dos utilizadores, de modo a encorajar a conversa. SHRDLU era capaz de analisar todos os nomes, verbos e regras simples de gramática. Tinha também alguma memória, mas apesar de pouca, conseguia ter contexto. Por exemplo, se no início da conversa falassem de um objeto vermelho, mais tarde o *chatbot* assumia essa cor em vez de outra. Weizenbaum achava possível ter progredido na área da inteligência artificial, mas foi tudo uma ilusão. Quando tentou alargar esta área, as regras necessárias eram inimagináveis. Após alguns anos, abandonou esta área e concluiu que era impossível dar às máquinas a verdadeira linguagem humana, tendo em conta as ferramentas disponíveis.

Em 1995, foi desenvolvido um *chatbot*, denominado por Alice (Lun, 2009), por Richard S. Wallace, na Lehigh University. Foi criado com recurso à linguagem AIML, que deriva da linguagem XML. É considerado como um *chatbot* de processamento de linguagem natural, que permite conversar com humanos, aplicando padrões heurísticos de reconhecimento de regras, segundo a entrada desses humanos. É um dos mais fortes programas deste tipo e ganhou o

⁴ Instruções através da linha de comandos.

Loebner Prize⁵, três vezes (2000, 2001 e 2004). Contudo, o programa não foi capaz de passar o teste de Turing⁶, porque muitas vezes ainda expunha aspetos mais mecânicos, mesmo em conversas curtas.

Em 2009, os investigadores da Universidade de Toronto mostraram que era possível reconhecer a fala com gravação de voz precisa e limpa.

Em 2010, surgiu a Siri (Apple, 2010), que é uma interface conversacional que usa processamento de linguagem natural de modo a responder perguntas, fazer recomendações, e executar ações. Encontra-se nos dispositivos iOS e tem vindo a evoluir ao longo dos anos.

Em 2012, o mesmo grupo de investigadores da Universidade de Toronto ganhou o concurso Machine Vision⁷, usando um algoritmo de *deep learning*⁸ que foi estrondosamente preciso.

Depois, e mesmo durante este período de tempo, foram criados outros tipos de assistentes de voz ou *chatbots*, mas atualmente os que têm mais destaque são: Amazon Echo (Amazon Developer, 2014), Google Allo (Google, 2016), Google Home (Google, 2016), IBM Watson (IBM, 2011), Microsoft Cortana (Microsoft, 2014) e Siri.

Desde 1966, algumas interfaces conversacionais foram então desenvolvidas, e na Figura 2 são apresentadas as que tiveram mais importância, começando pelo *chatbot* ELIZA, já referido anteriormente neste mesmo subcapítulo.

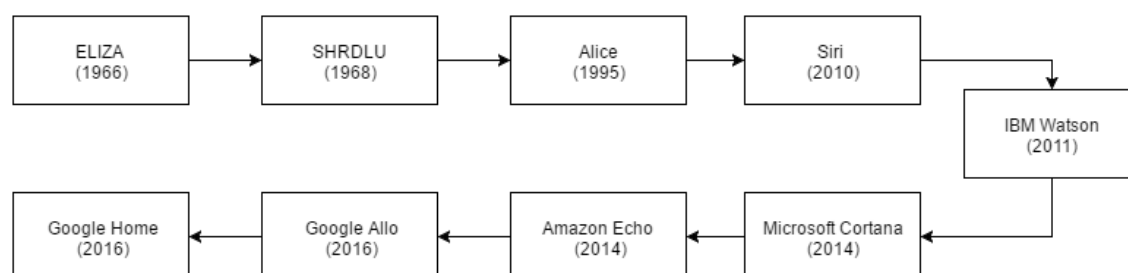


Figura 2 – Interfaces conversacionais com mais destaque desde 1966

É certo que estas não são as únicas interfaces conversacionais que apareceram desde 1966, mas são as que tiveram mais destaque e mais relevância ao longo da sua evolução.

⁵ Competição anual de inteligência artificial que direciona prémios aos *chatbots* que sejam considerados pelos jurados, os mais parecidos com humanos.

⁶ Teste que permite verificar a capacidade de uma máquina mostrar um comportamento semelhante a um ser humano.

⁷ Competição que existe desde 2012, para avaliar algoritmos de reconhecimento de imagem.

⁸ Ramo de *machine learning* baseado num conjunto de algoritmos que tentam modelar abstrações de alto nível em dados.

De acordo com este espaço de tempo, várias propriedades mudaram, e por isso existem algumas diferenças entre as interfaces conversacionais de antigamente, comparativamente às atuais. E são essas mesmas diferenças que são nomeadas na Tabela 6 (Brownlee, 2016).

Tabela 6 – Diferenças entre as interfaces conversacionais (antigas vs atuais)

| Tipos de Interfaces Conversacionais | Descrição |
|--|---|
| Antigas | <ol style="list-style-type: none"> 1. Uso de padrões e regras gramaticais. 2. Uso de comandos pré definidos. 3. As respostas eram muito limitadas. |
| Atuais | <ol style="list-style-type: none"> 1. Há processamento de linguagem natural, em vez de comandos. 2. Uso de redes neurais, por exemplo, que simulam o raciocínio do cérebro humano para estabelecer um diálogo mais próximo da linguagem natural. 3. Inteligência artificial cada vez mais evoluída, com recurso por exemplo a <i>deep learning</i>. Nesta geração já existem várias ferramentas para criação de <i>chatbots</i> com integração em aplicações de mensagens. |

Relativamente às interfaces conversacionais antigas, pode-se equiparar o *chatbot* ELIZA. Já quanto às atuais, pode-se associar a Siri e os assistentes de voz e *chatbots* referidos, que surgiram desde então.

Os processos usados ao longo deste período de tempo, iniciado em 1966 até agora, mudaram significativamente, e nos dias de hoje tem sido uma tarefa árdua conseguir evoluir ainda mais. Apesar dos computadores atualmente permitirem o processamento necessário para análise da conversa e conseguir dar respostas a um certo pedido efetuado pelo utilizador, tentar criar uma pessoa através de um agente conversacional é bastante difícil. As próprias tarefas de análise e processamento da conversa são muito trabalhosas e também os humanos são imprevisíveis.

2.1.2. Vantagens das Interfaces Conversacionais

Desde 1966 que as interfaces conversacionais foram introduzidas oficialmente, e se isso aconteceu foi porque foram encontradas vantagens associadas. Mas o mais importante é que se desde aí houve uma constante evolução ano após ano na área, é porque realmente essas vantagens não foram vistas em vão.

Este tipo de interfaces tem tido muita procura por parte de utilizadores, retalhistas e principalmente por empresas.

Atualmente, bastantes empresas ou mesmo projetos têm associados nos seus sítios *web*, uma componente de conversação, mesmo que seja em pequena quantidade. Esta componente aparece integrada nos serviços de *chat* que fornecem, de ajuda para os utilizadores. Normalmente numa fase inicial são compostas por um agente criado. Só numa fase mais à frente, é que a conversa passa a ser gerida por um humano. Uma vez que a fase inicial é sempre igual para todos os utilizadores, é fácil tornar esta tarefa automática, criando um *chatbot*. Um exemplo deste tipo de uso de *chatbots*, está inserido no sítio *web* da Microsoft (Figura 3).

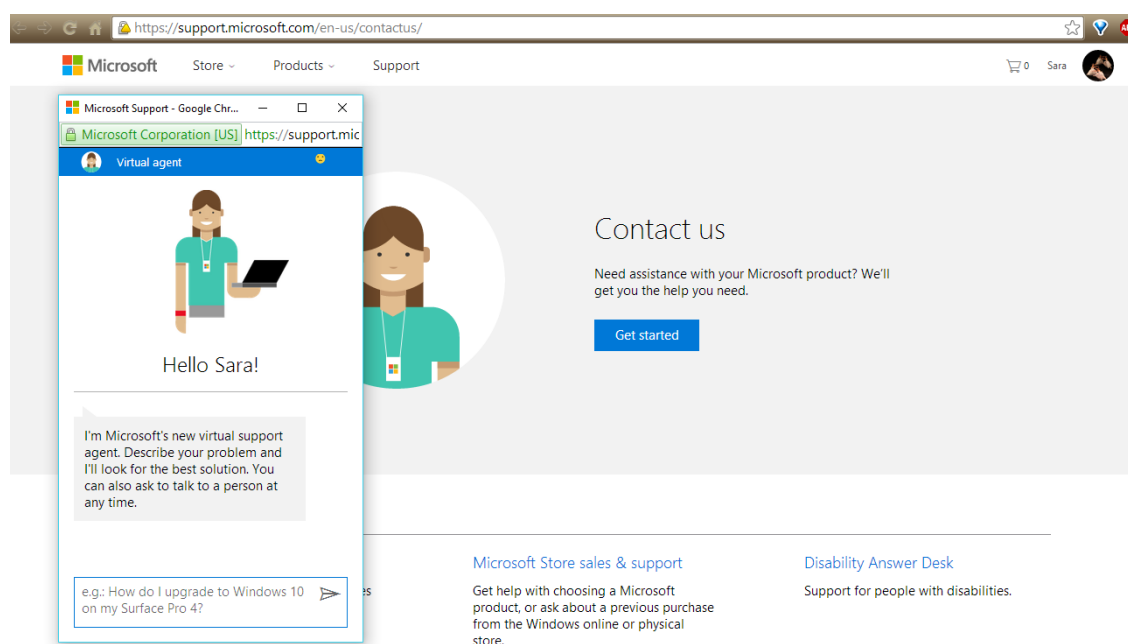


Figura 3 – Exemplo de *chatbot* (Microsoft, 2017)

Outro exemplo deste tipo de interfaces e das vantagens associadas, é apresentado através da empresa SMARKIO. De modo a converter mais visitantes em seguidores, experimentaram substituir formulários *web* dos seus clientes, por uma interface conversacional. Assim, concluíram que o envolvimento dos utilizadores foi bastante positivo e as taxas CTR de utilizadores que clicam para proceder para o passo seguinte, realmente aumentaram. Através do formulário esta taxa apresentava 25% de prosseguimento, enquanto que pelo uso de uma interface conversacional, 85%, ou seja, um aumento de 60%. Para além disso, os seguidores aumentaram em 200% (Rui Campos, 2017).

A multiplataforma é uma das razões pelas quais as interfaces conversacionais têm vindo a ser introduzidas nas empresas. Funcionam tanto em *smartphones*, computadores, *smartwatches*, e até mesmo em dispositivos sem ecrã, tal como o Amazon Echo (Figura 4). Para além disso, podem ser integradas em serviços tais como o Twitter, Facebook, Slack, entre outros (Brownlee, 2016).



Figura 4 – Amazon Echo

A maior vantagem é a possibilidade de os utilizadores encontrarem o que pretendem de um modo mais fácil e simples, sem terem de ser eles a tentarem descobrir. Para tal, apenas têm que falar através de voz ou não, com um agente que simula um humano. Esta é a maior dificuldade desta área, mas é também um ponto muito importante e desafiante. Os utilizadores terão um maior aproveitamento e um maior nível de experiência de utilizador, caso não se sintam a falar com um *robot*, mas sim com um humano.

2.1.3. Assistentes de Voz

Como o poder dos computadores cresce de ano para ano, também as limitações na interação entre o homem e a máquina vão diminuindo. Assim sendo, começa a ser cada vez mais possível, apesar de dificultosa essa tarefa, permitir uma comunicação mais comum para os humanos, ou seja, através de reconhecimento de fala (Rozmovits, 1996).

Os assistentes de voz são um tipo de interface conversacional, onde a conversa é controlada por fala. Um assistente utiliza representação lógica de relações, sequências e textos de linguagem humana para simular e reconhecer a fala humana e é baseado numa interação por áudio, incluindo elocuções e sons a produzir e a receber. Esta é controlada pelo computador ou máquinas que por sua vez são controladas pelo humano (Pinho, 2016).

A Siri da Apple e a Cortana da Microsoft são exemplos deste tipo de interfaces, disponíveis em dispositivos móveis e portáteis. Já o Amazon Echo e o Google Home são também outro tipo de exemplos, mas noutra formato. Estes são pequenos altifalantes com microfones muito

poderosos. Têm uma boa capacidade de audição mesmo com alguma distância entre o utilizador e o aparelho, e conseguem entender a tarefa desejada em milissegundos.

Amazon Echo e Google Home são a personificação de assistentes de voz. Basta oferecer-lhes um local na casa no qual seja possível interagir com eles ao longo do dia (Williams, 2016).

2.1.4. Chatbots

Chatbots são programas de computador que imitam conversas com pessoas, usando inteligência artificial. Eles podem transformar a forma como se interage com a internet de modo a se obter tarefas auto iniciadas. Essas tarefas passam a ser conseguidas através de uma conversa (Wong, 2016) textual, sem voz. O seu maior objetivo é tentar automatizar algo, através de uma conversa por *chat*.

Os melhores exemplos de aplicações de mensagens são o Facebook Messenger e Whatsapp, mas, de facto, essas e outras plataformas têm vindo a investir em *robots*, tais como o Slack e o Twitter (An, 2016).

Muitas empresas estão a investir na criação de *chatbots* para integração no Facebook Messenger, usando plataformas que permitem esse desenvolvimento (essas ferramentas serão melhor exploradas no capítulo 2.3). Isto porque a sociedade hoje em dia está muito ligada às redes sociais e ao que estas fornecem. A plataforma do Facebook Messenger é realmente muito utilizada e é um bom meio de comunicação através de mensagens, com uma forte influência. Assim sendo, têm vindo a aparecer cada vez mais *robots* associados a empresas, para serem utilizados via Messenger. Como à partida o Facebook Messenger já está instalado nos *smartphones*, não é necessário procurar aplicações relativas aos *chatbots* que os utilizadores pretendem utilizar, nas lojas de aplicações, nem aguardar pela sua instalação. Estes já se encontram no Messenger, tornando-se num processo mais simples e rápido. Já que é um meio tão simples e tão utilizado, juntou-se o mundo dos *chatbots* a esta plataforma

Outras empresas investem muito também noutro tipo de *chatbots* que são os *chatbots* inseridos em sítios *web*. Estes têm o objetivo de dar suporte aos utilizadores, como é o caso da Microsoft (Figura 3).

2.1.5. Síntese do Subcapítulo

Neste subcapítulo das Interfaces Conversacionais foi feita uma apresentação mais evolutiva do tema, desde 1966 até hoje. Desde o primeiro *chatbot* (ELIZA), que a área das interfaces conversacionais tem vindo a evoluir muito. Apesar de trabalhosa esta tarefa, tem sido possível aproximar a conversa entre o agente e o utilizador, de uma forma muito mais natural e semelhante a uma conversa entre dois humanos.

Para além disso, foram nomeadas as vantagens existentes relativamente ao uso de interfaces conversacionais, em Vantagens das Interfaces Conversacionais. Uma interface deste tipo é muito vantajosa em várias áreas e se continuam a investir na sua evolução, é porque realmente é algo que vale esse esforço.

Por fim, após esta visão detalhada nos subcapítulos referidos anteriormente, foi realizada uma comparação entre os assistentes de voz e os *chatbots*, que são os dois tipos de interfaces conversacionais existentes. No primeiro tipo, a conversa é realizada por voz, enquanto que no segundo, a conversa é concretizada textualmente. Em cada um deles foram também apresentados alguns exemplos.

Por exemplo, depois de testado o Amazon Echo, e fazendo uma comparação relativamente ao Google Home, verificam-se algumas diferenças significativas e que a nível de experiência de utilizador, definem realmente uma preferência (Tabela 7 e Tabela 8).

Tabela 7 – Tabela de comparação entre Amazon Echo e Google Home (I)

| Detalhe | Amazon Echo | Google Home |
|--|--|--|
| Tempo de espera de ligação | Mais lento. | Mais rápido. |
| Mudança de voz do agente, consoante o tema a falar | Não. | Sim. |
| Tipo de conversa | É um pouco limitada. A Alexa (agente do Amazon Echo) praticamente só responde ao que utilizador pede, e muitas vezes nem reconhece o que lhe foi dito. | Há mais diálogo, não só partindo do utilizador. O Google Home interage com o utilizador tendo em conta algum tempo de pausa da parte do utilizador. |
| Apresenta detalhes visuais | Sim. | Sim. |
| Tipos de detalhes visuais | Apresenta uma luz azul que indica de que lado vem a voz do utilizador, e que simultaneamente indica que está ativo. Tem também uma luz vermelha que indica que não está ativo. | Apresenta luzes que indicam que está a ouvir, o volume do som, alguns detalhes relativos ao que é apresentado/ouvido. Por exemplo, se for pedida uma música de natal, mostra uma coroa típica de natal com luzes vermelhas e verdes. |

Tabela 8 – Tabela de comparação entre Amazon Echo e Google Home (II)

| Detalhe | Amazon Echo | Google Home |
|------------|---|--|
| Tem botões | Sim. Um que permite ligar/desligar o microfone e outro relativo à ação. | Sim. Apenas um que permite ligar/desligar o microfone. As restantes ações são controladas por toque, no topo do dispositivo. |

De um ponto de vista mais pessoal, seria escolhido o Google Home porque não se sente uma relação tão pessoa-*robot*. Há um tipo de interação que não existe no Amazon Echo, e que se sente falta, que é um tipo de interação mais pessoal. Há certos detalhes que este último não entende, enquanto o primeiro para além de entender, ainda permite escolher certas opções relativas ao pedido feito.

Por exemplo, se forem pedidas as notícias do dia, sempre que o tema muda, também muda a voz (passa de uma feminina para masculina, ou então de uma feminina para outra), e nenhuma delas parece robótica. Apesar de a voz da Alexa parecer um pouco mais semelhante à de um *robot*, é também um dispositivo muito útil que tenta simular um humano, mas com um pouco mais de dificuldades. É normal que hajam estas diferenças, também porque o Amazon Echo surgiu dois anos antes do Google Home. Apesar de haverem constantes atualizações, é possível verificar algumas diferenças notórias.

Ambos mostram as vantagens do uso de interfaces conversacionais e como podem simplificar a vida das pessoas. Tal como referido no capítulo 2.1.3, ambos são a personificação de assistentes de voz.

2.2. Trabalho Relacionado

Apesar da área de interfaces conversacionais estar a crescer cada vez mais, e ultimamente terem sido desenvolvidos bastantes assistentes de voz ou *chatbots*, ainda não existe nada semelhante para a área de cultura, artes e entretenimento, como é o caso de onde a Casa da Música se insere. Contudo, existem alguns trabalhos já desenvolvidos, que de certa forma se assemelham ao que é pretendido criar para esta instituição.

As interfaces conversacionais podem ser inseridas em vários contextos e áreas, tal como é possível constatar através dos exemplos a serem abordados nos subcapítulos seguintes. Estes exemplos são interfaces já criadas, em que a conversação tem algum objetivo final: compra de algo, visualização de conteúdos, entre outros.

Todos estes *chatbots* foram desenvolvidos para uso através do Facebook Messenger, tal como pretendido no âmbito desta dissertação.

Os subcapítulos seguintes falarão dos seguintes *chatbots*: 1-800 Flowers, CNN, Hi Poncho e KLM. No final será feita uma síntese de toda esta análise de trabalho relacionado.

2.2.1. 1-800 Flowers

1-800 Flowers é uma loja florista a retalho, criada por Jim McCann em 1976.

Em 2016 foi criado um *chatbot* associado a esta marca, para que as pessoas que comprem os seus produtos, possam fazê-lo mais facilmente, através do Facebook Messenger (An, 2016). Este *chatbot* simples, responde a pedidos vindos do utilizador, em forma de texto e mostra um carrossel de produtos para que o utilizador possa escolher a opção que pretende.

No início da conversa, o agente dá ao utilizador duas opções: comprar flores ou falar com o suporte (Figura 5).

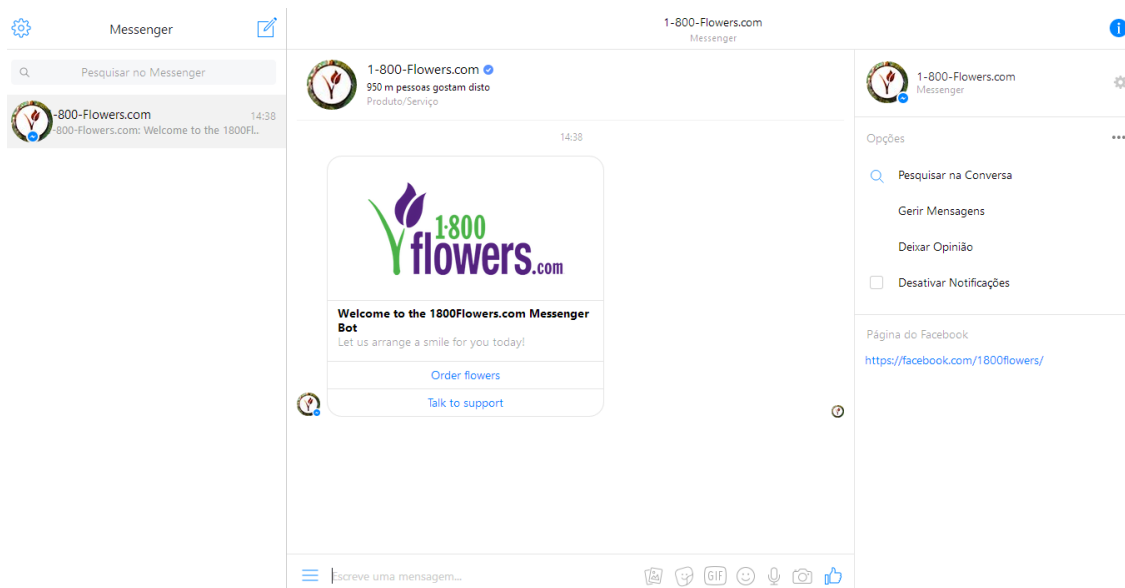


Figura 5 – Conversa iniciada com o 1-800 Flowers (Facebook Messenger)

Caso o utilizador selecione a primeira opção, a primeira coisa que é pedida, é a morada de destino das flores (Figura 6).

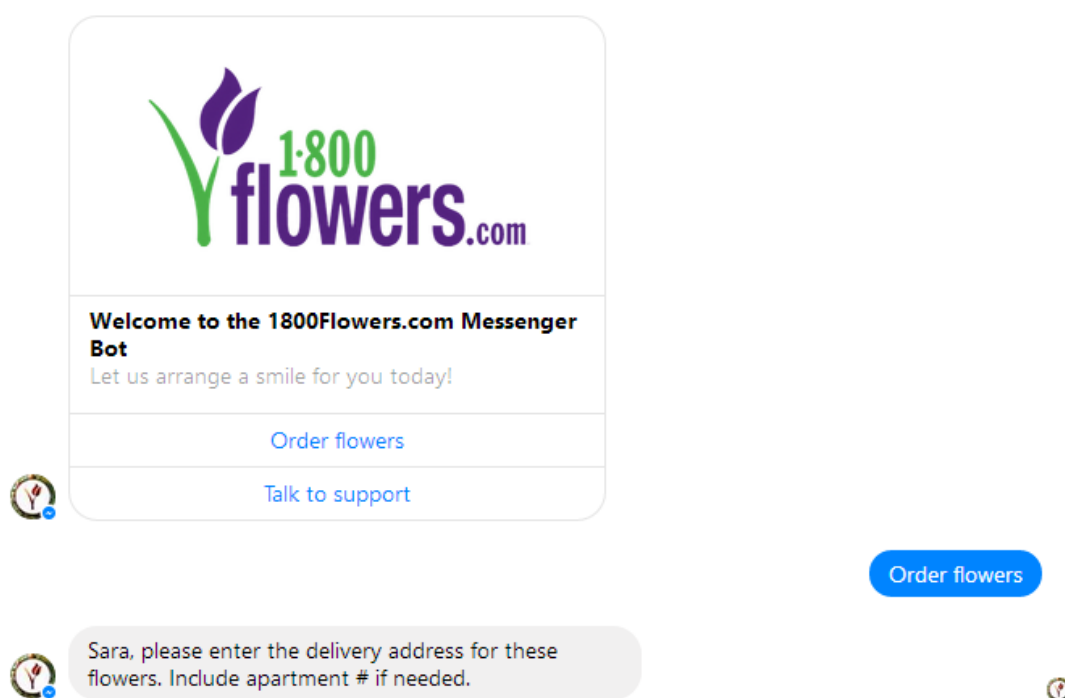


Figura 6 – Ação de compra de flores (1-800 Flowers)

Após essa indicação, é necessário referir qual a categoria pretendida, assim como o ramo de flores que se deseja. Os últimos passos são selecionar uma data de entrega, e inserção de mais alguns dados do recetor, assim como pessoais (Messenger Chatbots, 2016).

2.2.2. CNN

CNN é conhecida por ser a fonte mais segura e confiável, no que diz respeito a notícias e informações.

Tal como no exemplo anterior, em 2016 foi também criado um *chatbot* para dar ao utilizador, notícias de acordo com a categoria que pretendem. Assim é só indicarem essa mesma categoria, que o agente direciona para a conversa criada, todas as notícias do dia, relacionadas com esse tema. É criado mais uma vez um carrossel com todas as notícias e em cada uma estão associadas duas opções de escolha: ler a notícia, ou algo mais (Figura 7).

24 - Estado da Arte

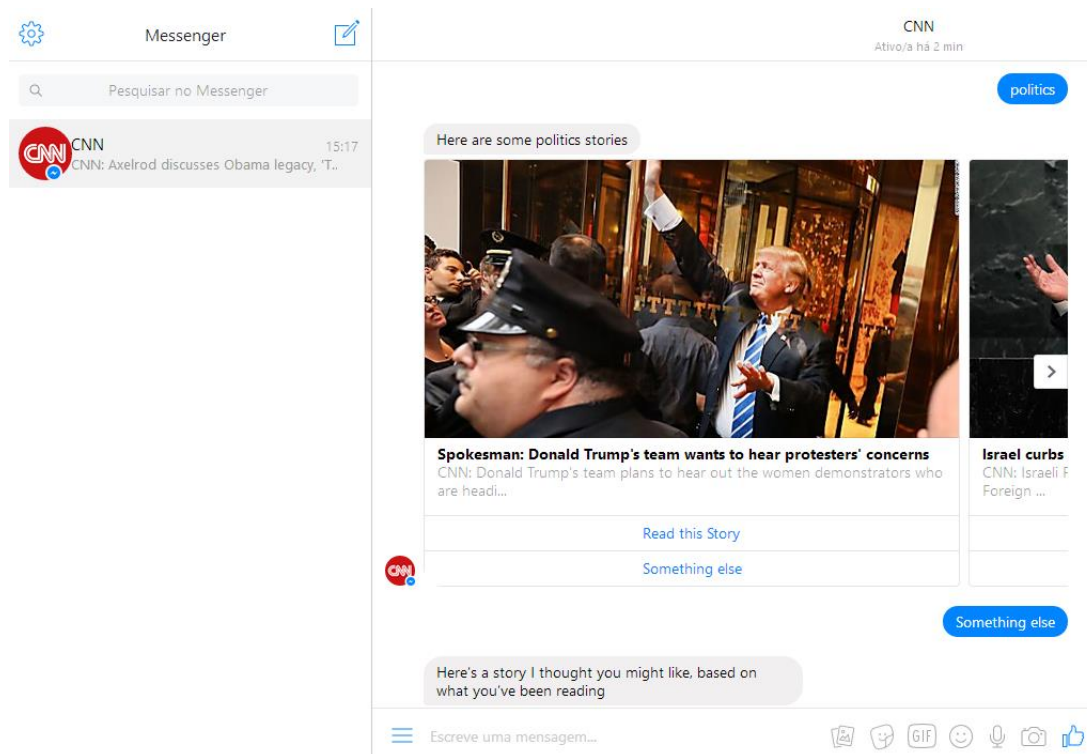


Figura 7 - Conversa iniciada com a CNN (Facebook Messenger)

A primeira opção abre a notícia diretamente no sítio *web*, enquanto que a segunda faz com que o agente envie para o utilizador outras notícias que ele talvez goste, baseadas na notícia através da qual se selecionou aquela opção (Figura 8).

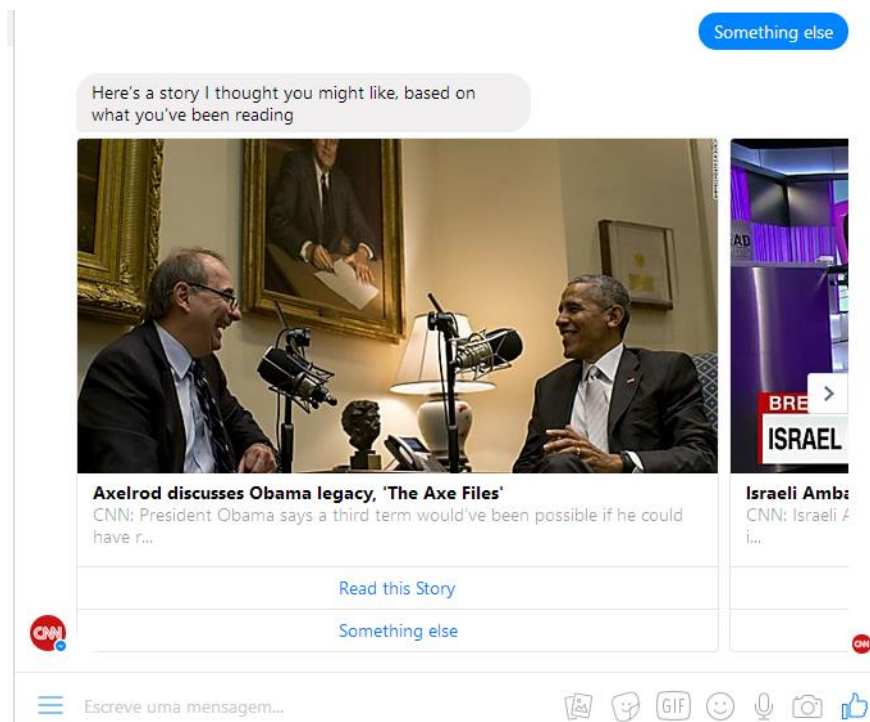


Figura 8 - Ação de seleção da opção: Algo mais (CNN)

2.2.3. Hi Poncho

Poncho é um *chatbot* que tem como objetivo enviar para os utilizadores, previsão meteorológica, horóscopo, entre outros. O agente é representado por um gato, e tem personalidade (botlist, 2016), que é algo bastante importante para que os agentes não se tornem apenas *robots*, mas sim, o mais parecido com humanos. Tal como os outros, este *chatbot* foi também criado em 2016.

Ao iniciar uma conversa com o Poncho, este necessita do local desejado para receção da previsão meteorológica (Figura 9). Pode ser indicado, pesquisando pela localização, segundo uma opção existente (*Send Location*) com tal objetivo, ou escrita.

26 - Estado da Arte

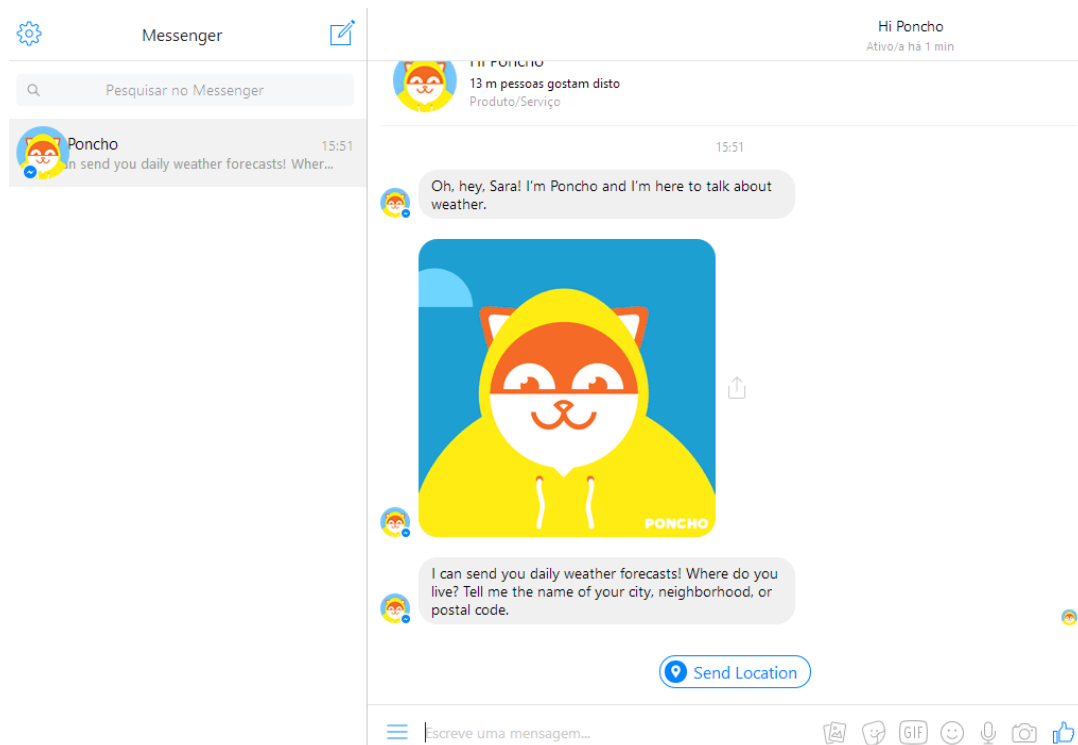


Figura 9 - Conversa iniciada com o Hi Poncho (Facebook Messenger)

Ao escolher essa opção de envio de localização, o agente pede confirmação da parte do utilizador, para saber se a localização está correta. Tendo esta confirmação, é enviada de seguida a previsão meteorológica, que como se pode constatar, existe personalidade do lado do agente (Figura 10).

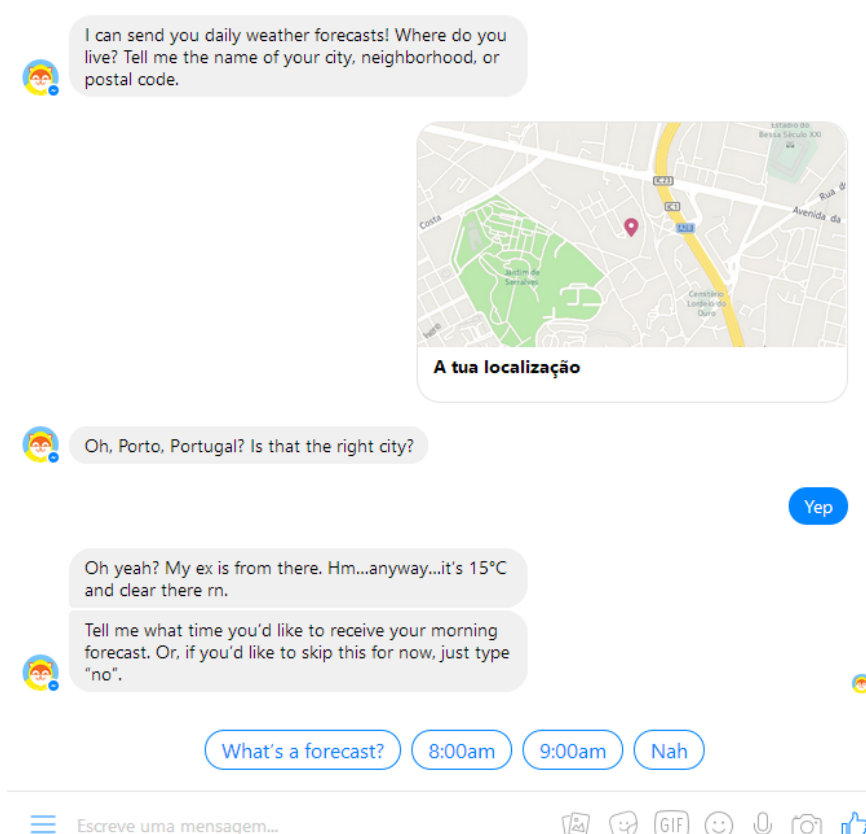


Figura 10 - Ação de verificação de localização (Hi Poncho)

Após esta ação, é possível executar outro tipo de ações, tais como seleção de um horário para receção da previsão meteorológica, pela manhã, entre outras.

2.2.4. KLM

KLM é uma empresa holandesa de transporte aéreo. A partir de 2016, passou a disponibilizar um novo serviço (KLM, 2016): receção da documentação do voo, via Messenger, ou seja, um novo *chatbot*.

Depois de agendar os voos no sítio *web* da empresa, é possível escolher receber a confirmação de reserva, notificação de *check-in*, cartão de embarque e atualizações de estado do voo, via Messenger. Isto torna a informação mais fácil de se encontrar, uma vez que se encontra num único local, disponível no aeroporto, em deslocações, ou em casa. É possível também fazer qualquer tipo de questão, através do Messenger.

Após o utilizador ter feito uma compra através do sítio *web* do KLM, este escolheu receber a confirmação da reserva, assim como notificações relativamente ao *check-in* (Figura 11).

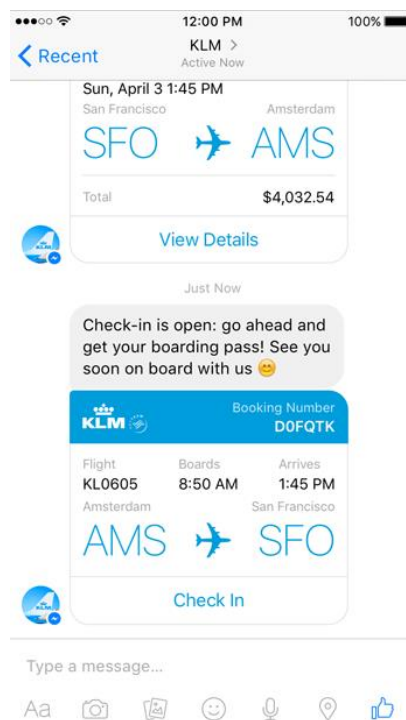


Figura 11 – Confirmação de reserva e notificações de *check-in* (KLM)

E após o utilizador ter feito o *check-in* através da página pessoal do mesmo, no KLM, seleccionou a opção de receção do estado do voo, mais uma vez através do Messenger (Figura 12).

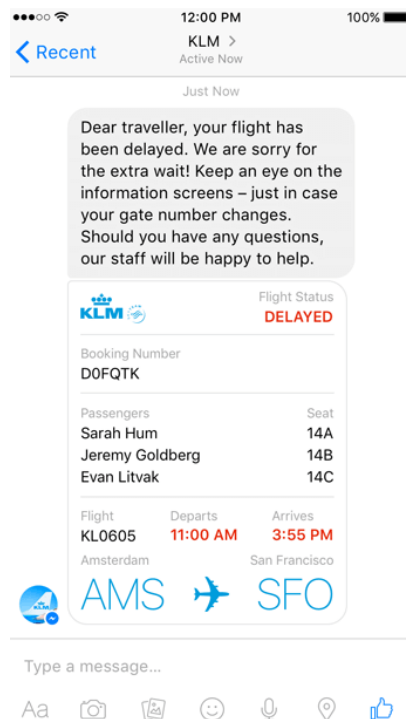


Figura 12 – Receção de atualizações do estado do voo (KLM)

2.2.5. Síntese do Subcapítulo

Neste subcapítulo foram listados alguns exemplos de trabalho já existentes e relacionados de certa forma com o que é pretendido desenvolver no âmbito desta dissertação. Estes *chatbots* que serviram de exemplo, foram todos desenvolvidos para serem utilizados através do Facebook Messenger.

No ano de 2016 surgiram variados *chatbots* no Facebook Messenger, e estes são só alguns exemplos. São estes os que mais se assemelham com o protótipo que se tem como objetivo desenvolver: 1-800 Flowers, CNN, Hi Poncho, KLM. Neste mesmo ano surgiram do mesmo modo, variadas interfaces conversacionais que não têm por obrigatoriedade ligação com o Facebook Messenger (por exemplo: Google Allo, Google Home, entre outros). E não só vários *chatbots* foram criados neste ano, mas também ferramentas que permitem o seu desenvolvimento (subcapítulo 2.3).

Realmente 2016 foi um ano de bastante desenvolvimento nesta área das interfaces conversacionais e representa o futuro em várias empresas.

2.3. Ferramentas para Criação de Sistemas de Diálogo

Atualmente existem várias ferramentas que têm vindo a ser desenvolvidas, com o objetivo de permitir criar agentes de conversação. Estes agentes têm associados diversas plataformas de integração, tais como Facebook Messenger, Slack, Twitter, Skype, Amazon Alexa, Microsoft Cortana, entre outros.

De seguida destacam-se algumas ferramentas livres e gratuitas que são atualmente bastante utilizadas em várias áreas de negócio. Estas ferramentas estão mais direcionadas para a integração com o Facebook Messenger, uma vez que é um dos objetivos desta dissertação.

A primeira ferramenta a ser explorada será a Wit.ai, seguindo-se da Api.ai. Após estas, será abordada a ferramenta Chatfuel. Em cada uma delas será realizada uma breve apresentação, assim como referido o seu funcionamento.

2.3.1. Wit.ai

Wit.ai (Wit.ai, 2016) é um serviço que fornece uma boa combinação entre o reconhecimento de voz e *machine learning* para desenvolvedores. Permite a conversão de comandos verbais em texto, e pode ser treinado para aprender esses comandos.

Em 2015 juntou-se ao Facebook e abriram uma plataforma completamente gratuita para instâncias públicas e privadas também. O seu desenvolvimento tem sido contínuo desde aí, com

novos recursos que têm vindo a aparecer, numa época em que não há nenhum sinal de abrandamento.

Wit.ai tem dois elementos principais – *intents* e *entities*. O primeiro representa que ação uma instrução pode realizar (ex. Ligar a luz). Já o segundo é um objeto específico ou parte de informação que a inteligência artificial precisa de saber para executar esse *intent* (ex. Qual luz? É uma luz inteligente?). Em vez de ser necessário criar *intents* de raiz, wit.ai providencia acesso a *intents* existentes através da comunidade de desenvolvedores.

Tem também associado o conceito de *roles*, onde se pode aprender a diferenciar entidades em contextos diferentes (ex. números em diferentes partes de uma instrução, podem-se referir a diferentes temas – tais como idade, encomenda, contagem). Tem ainda alguns tipos de entidades construídas, de modo a entender URLs, *emails*, duração, temperatura, entre outros.

Está disponível uma API para desenvolvedores de iOS, Android, Node.js, Raspberry Pi, Ruby, Python, C, Rust e Windows Phone. Há também um *plugin*⁹ em Javascript para desenvolvedores *front end* (Catanzariti, 2015). É contudo uma ferramenta direcionada para integração com o Facebook Messenger.

Na Figura 13 é apresentada a página pessoal relativa ao utilizador com sessão iniciada, apresentando as aplicações já existentes.

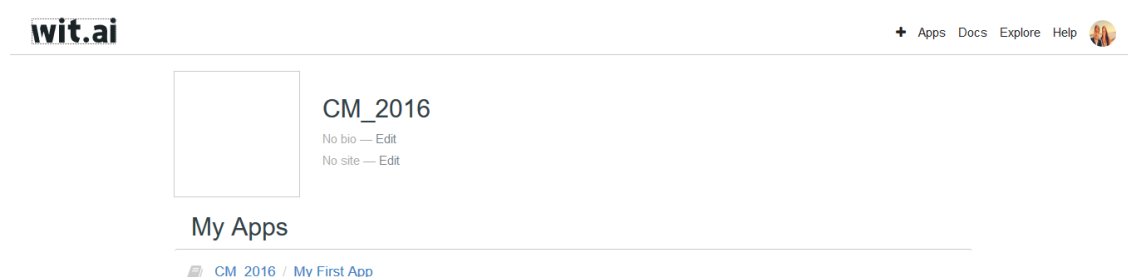


Figura 13 – Página pessoal no Wit.ai (aplicações já existentes)

Abrindo a única aplicação existente para este utilizador, é possível verificar quais as histórias existentes, ou seja, os casos de uso que podem ocorrer (Figura 14).

⁹ Módulo usado para adicionar funções a programas maiores, provendo alguma funcionalidade especial ou muito específica.

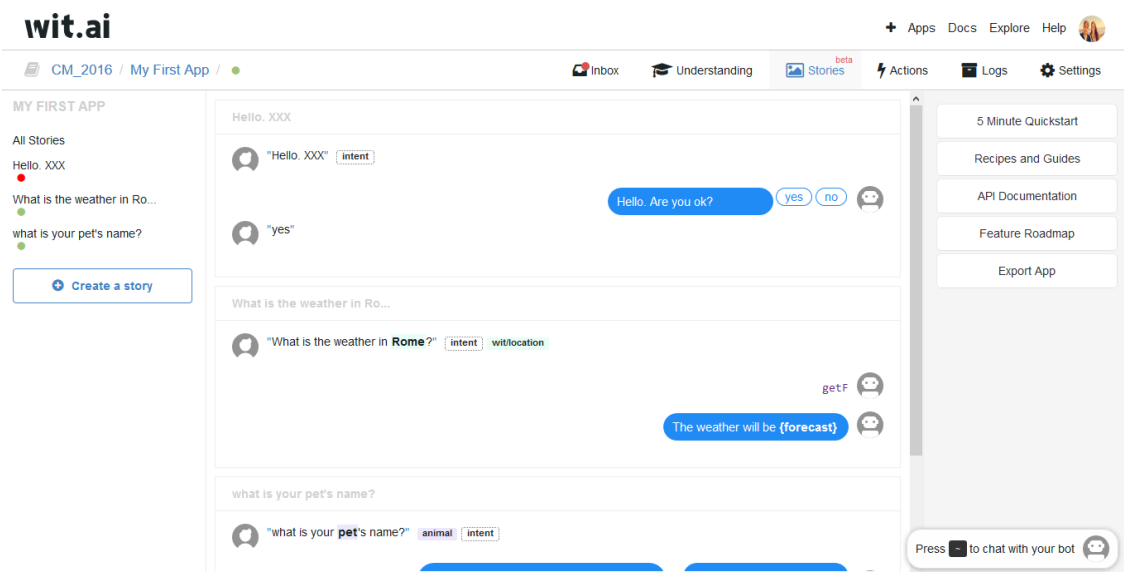


Figura 14 – Página associada a um *chatbot* - Histórias (Wit.ai)

Criando uma história nova, é a página da Figura 15 que é apresentada. É possível definir aquilo que o utilizador poderá dizer, assim como o que o agente realizará.

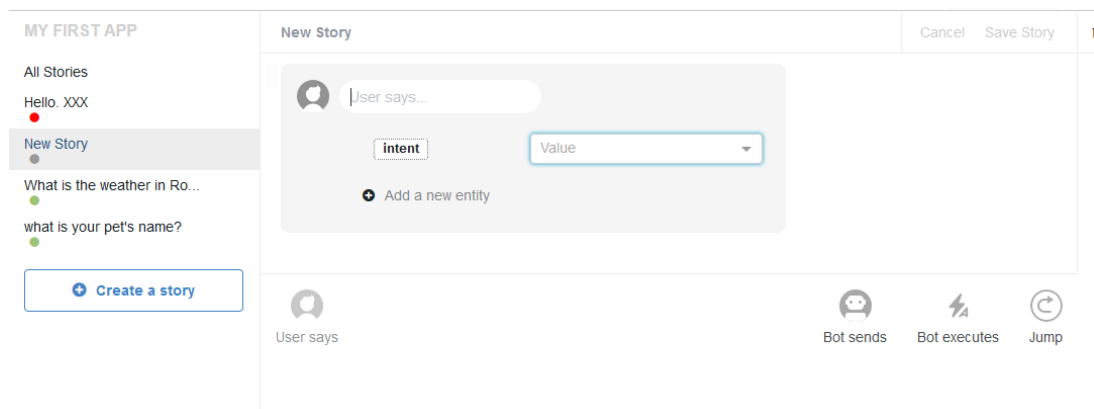


Figura 15 – Criação de uma nova história (Wit.ai)

Por último na Figura 16 é apresentada a página de conhecimento de entidades associadas a este *chatbot*. É possível aqui, verificar como o agente entende uma frase escrita, ou seja, testar se o seu funcionamento é bom ou não.

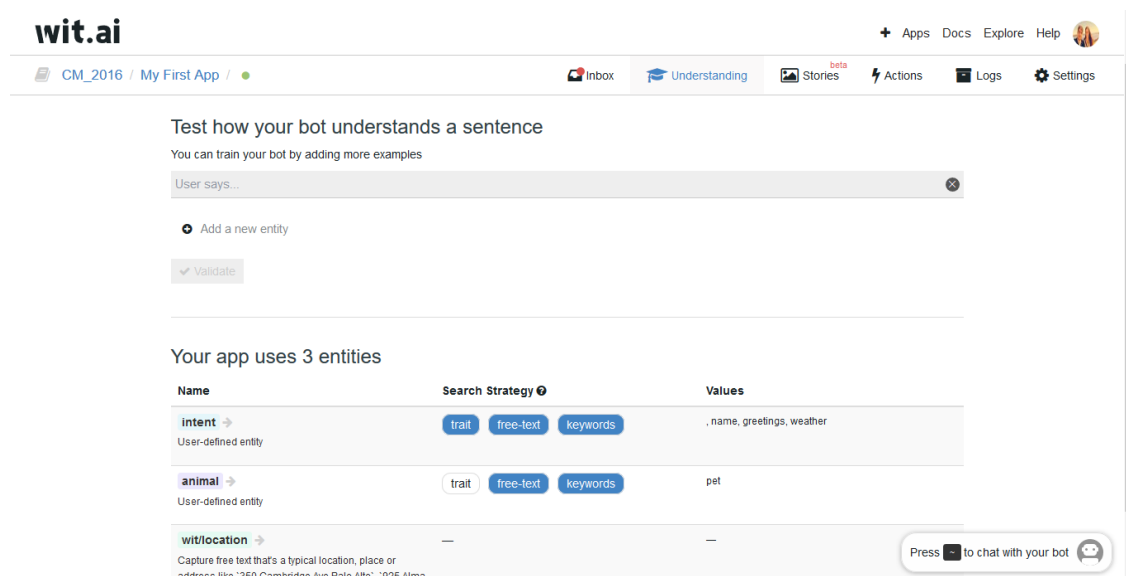


Figura 16 – Página do Wit.ai relativa ao conhecimento de entidades

2.3.2. Api.ai

Api.ai (Api.ai, 2016) que pertence à Google, é outro serviço que providencia capacidades similares ao Wit.ai, com *intents* e *entities*. Tal como Wit.ai, Api.ai é gratuita para utilização. Contudo tem uma opção paga que permite que este execute internamente numa *cloud* privada.

Um ponto-chave da Api.ai que difere de Wit.ai, são os seus *domains*. Estes domínios são uma coleção de conhecimento e dados estruturados da Api.ai que estão prontos para usar em qualquer agente. Estes agentes são as aplicações criadas através desta ferramenta. Os domínios podem incluir o conhecimento de verbos comuns e tipos de contexto. Contém variada informação real sobre tópicos de enciclopédia tais como história, definições de palavras, significância de certas pessoas (celebridades, escritores, personagens, entre outros), filmes, e muito mais.

Api.ai tem *SDKs* para Android, Apple Watch, Node.js, Cordova, Unity, C#, Xamarin, Windows Phone, Python e Javascript. Pode ainda ser integrada com o Amazon's Echo e Microsoft's Cortana (Catanzariti, 2015), mas tem no total catorze tipos de integração com serviços.

Após autenticação no sítio *web* desta ferramenta, a página inicial é a apresentada através da Figura 17. É aberto logo o agente onde se trabalhou em último lugar, que neste caso é o *Weather*.

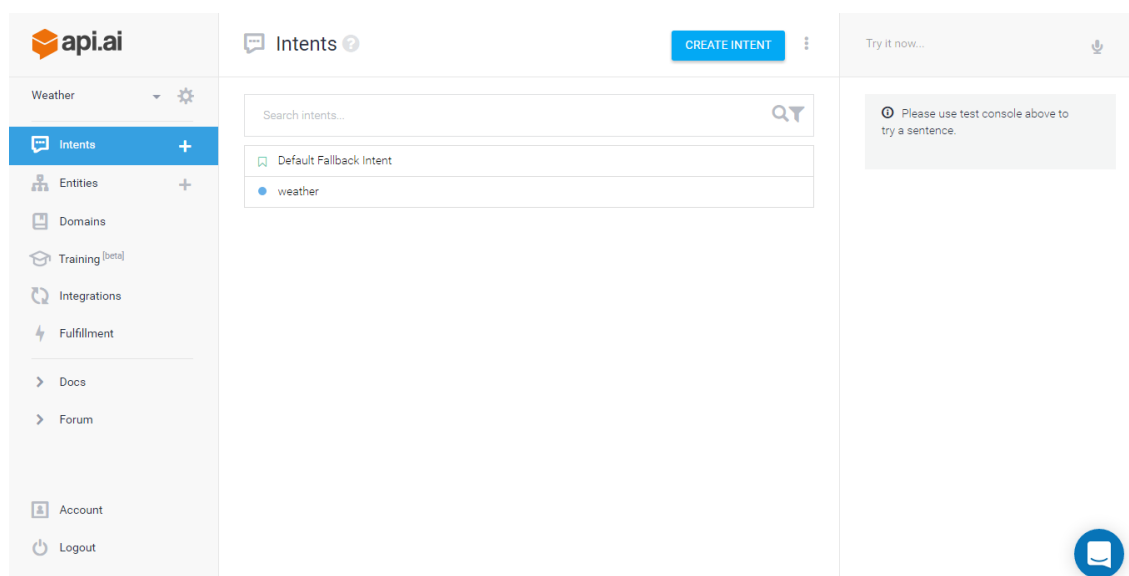


Figura 17 – Página inicial da Api.ai após autenticação

É possível visualizar todos os agentes criados através da barra lateral esquerda (Figura 18).

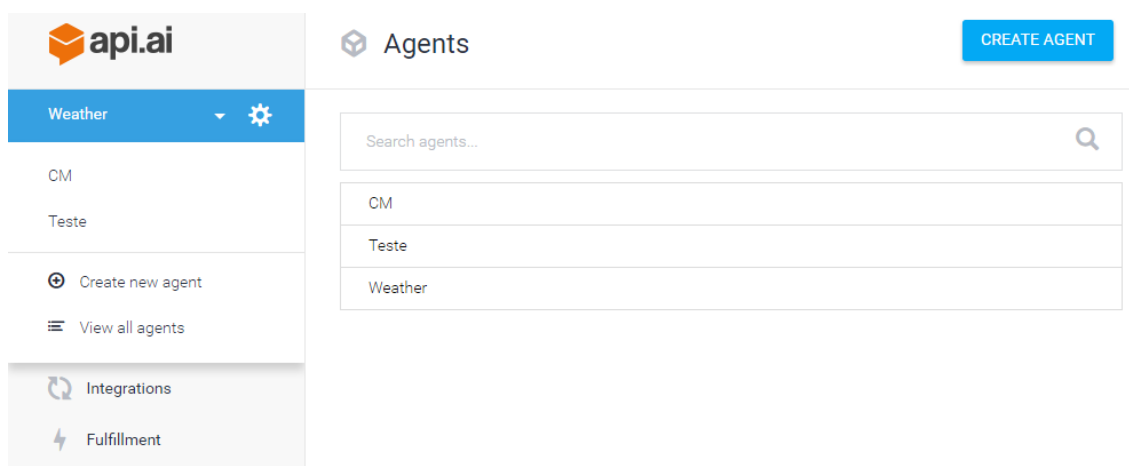


Figura 18 – Página dos agentes criados (Api.ai)

Cada agente tem associada uma página para visualização dos *intents* relacionados. Aqui são definidas as entidades para frases que o utilizador pode dizer, assim como respostas da parte do agente (Figura 19).

The screenshot shows the Api.ai console interface. On the left is a sidebar with navigation options: Weather, Intents (selected), Entities, Domains, Training [beta], Integrations, Fulfillment, Docs, Forum, Account, and Logout. The main area is titled 'weather' and includes a 'SAVE' button. It contains several sections: 'Add user expression' with a text input containing 'weather in London'; 'Events' section; 'Action' section with a dropdown set to 'yahooWeatherForecast'; a table for parameters; and a 'Response' section with a 'Text response' input.

| REQUIRED | PARAMETER NAME | ENTITY | VALUE | IS LIST |
|--------------------------|----------------|-----------------|----------------|--------------------------|
| <input type="checkbox"/> | geo-city | @sys.geo-city | \$geo-city | <input type="checkbox"/> |
| <input type="checkbox"/> | Enter name... | Enter entity... | Enter value... | <input type="checkbox"/> |

Below the table is a '+ New parameter' link. The 'Response' section shows a 'Text response' input with a sample response: '1 It seems the weather service is unavailable right now'.

Figura 19 – Página de um *intent* criado (Api.ai)

Para cada agente, estão também associadas as entidades (*entities*) referentes. Podem definir-se módulos, que podem ter variados valores com respectivos sinónimos (Figura 20).

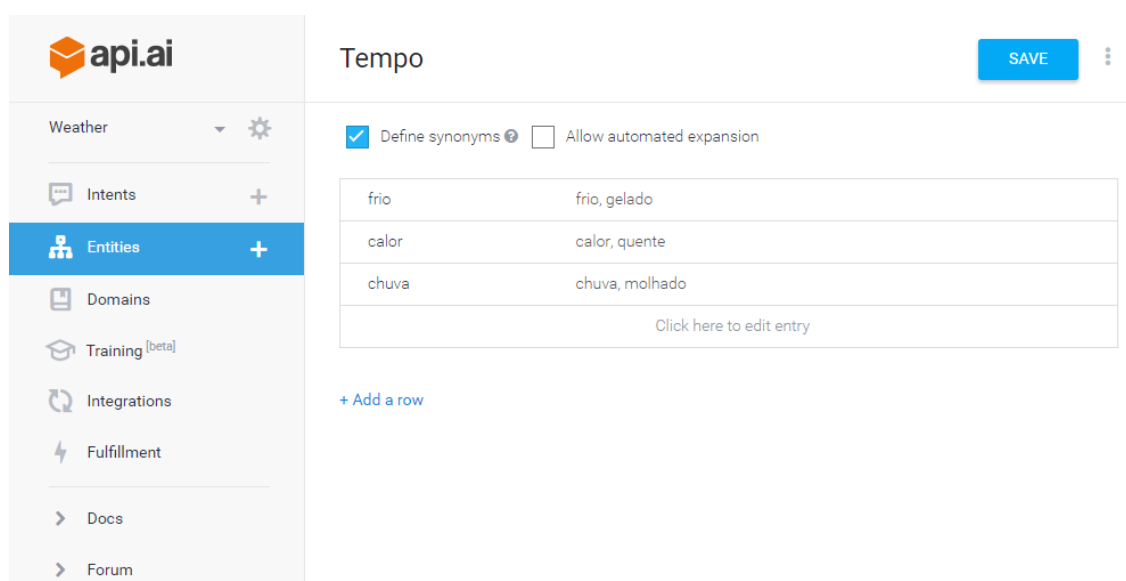


Figura 20 – Página de uma entidade criada (Api.ai)

Outro conceito aqui presente são os domínios (Figura 21). Há vários predefinidos, que podem ser associados ao agente. Por exemplo, se um projeto estiver relacionado com compras, pode ser adicionado esse domínio.

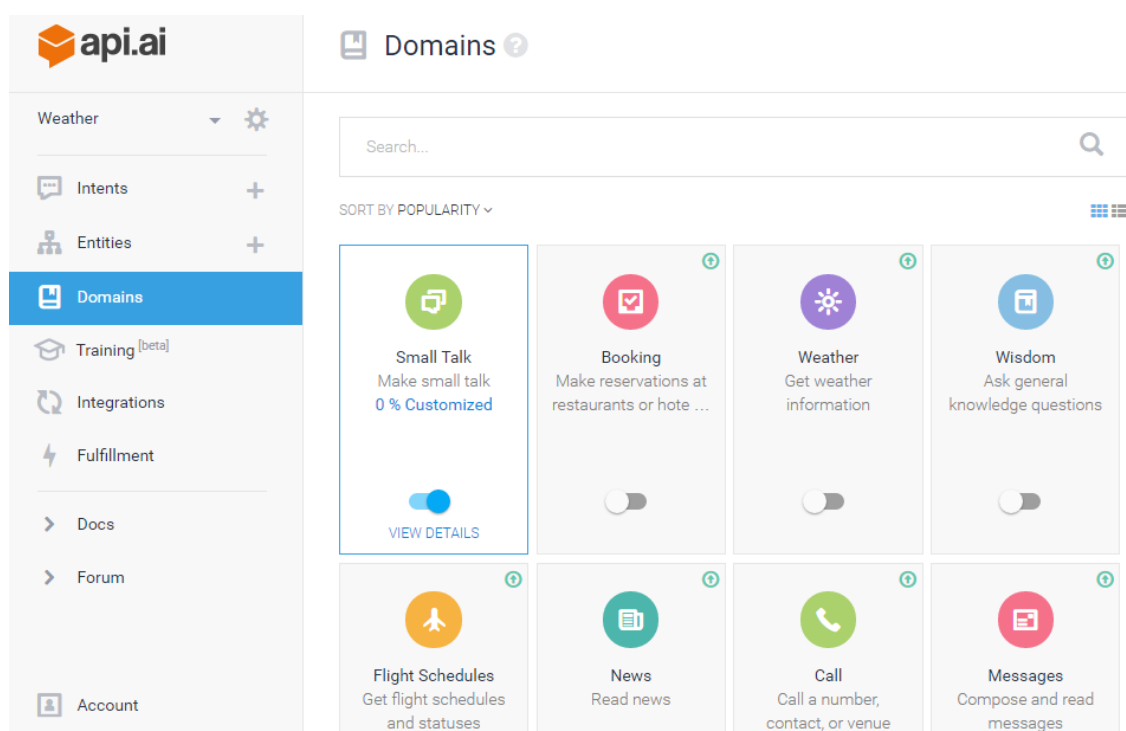


Figura 21 – Página dos domínios do chatbot (Api.ai)

Por fim, referente a esta ferramenta, é apresentada a páginas das integrações que a Api.ai disponibiliza (Figura 22). São catorze essas integrações.

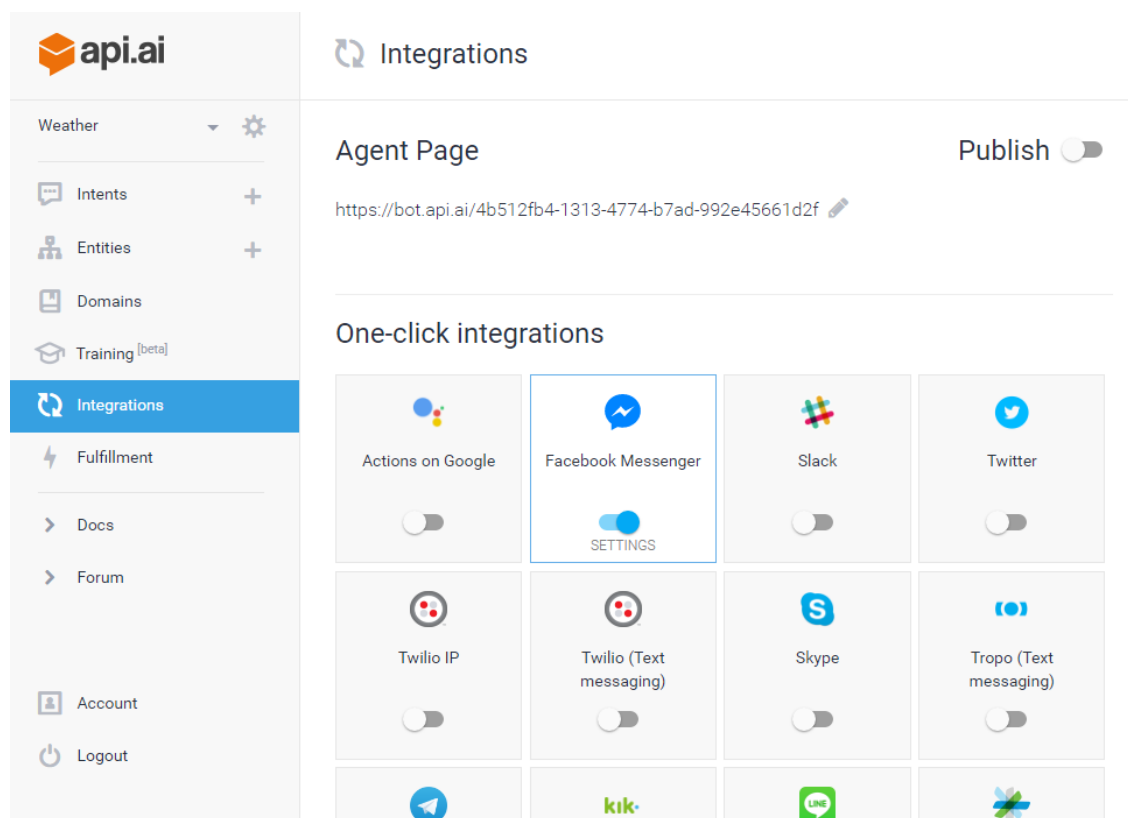


Figura 22 – Página das integrações associadas ao *chatbot* (Api.ai)

2.3.3. Chatfuel

Chatfuel (Chatfuel, 2015) nasceu em 2015, com o objetivo de construir um meio de construção de *robots*, para qualquer pessoa. Começou com o Telegram e rapidamente cresceu para milhões de utilizadores. Atualmente, o foco está na possibilidade de permitir a todas as pessoas desenvolverem *chatbots* para o Facebook Messenger.

O Chatfuel é uma ferramenta que permite criar *chatbots* para Facebook Messenger e Telegram, tendo assim dois tipos de integrações diretas com serviços. Para tal, não são necessárias competências de programação, para além de que é possível editar as respostas, assim como adicionar hiperligações (Agus, 2016).

Esta ferramenta foi utilizada pela Uber, Denver Broncos, National Geographic, Sacramento Kings, Tech Crunch e Complex.

A Figura 23 apresenta a interface do sítio *web* do Chatfuel, tendo como seleção, a página do *dashboard*. Esta página apresenta os *chatbots* criados, tendo em conta a conta com sessão iniciada.

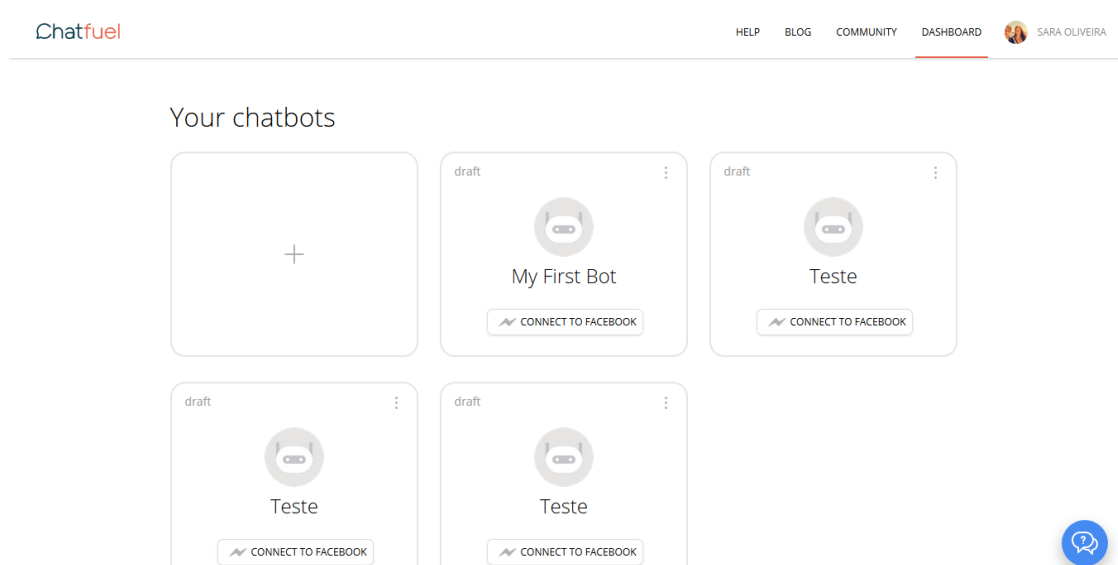


Figura 23 – Dashboard do Chatfuel

Selecionando um dos *chatbots*, abre a página apresentada na Figura 24, já com uma mensagem predefinida para as boas vindas. Esta página é que permite a criação do diálogo entre o agente e o utilizador, para o Facebook Messenger.

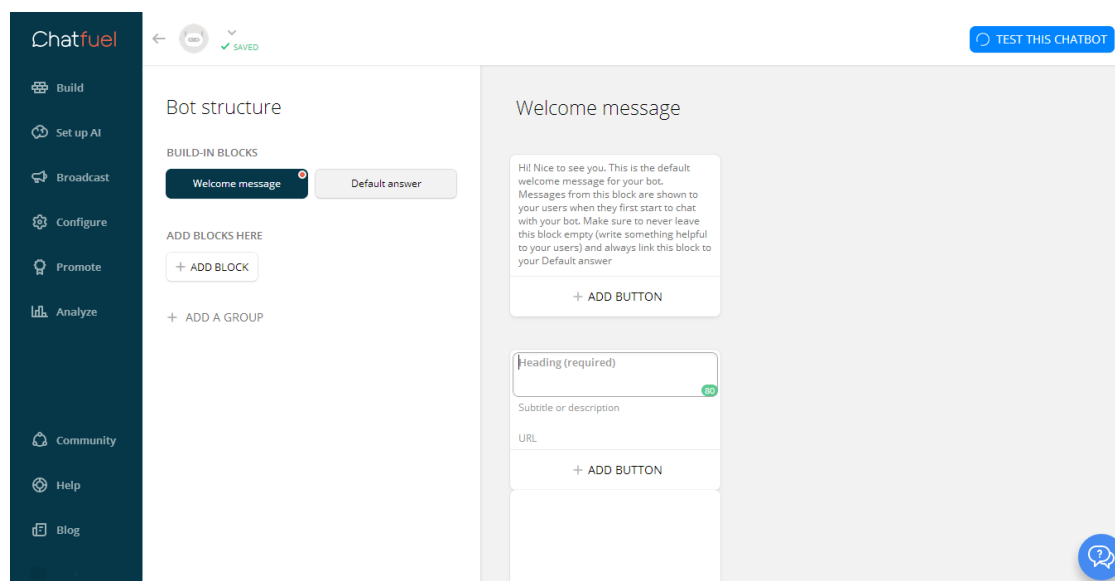


Figura 24 – Página de definição da conversa associada a um robot (Chatfuel)

Na secção seleccionada na Figura 24, foi definida uma mensagem de exemplo. Essa mensagem contém uma imagem, texto e um botão com direccionamento para um URL.

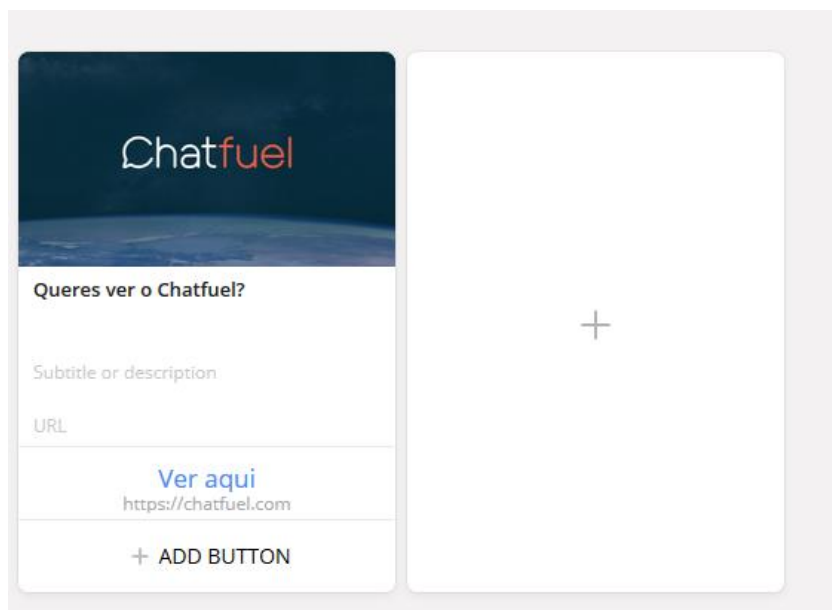


Figura 25 – Definição de uma mensagem (Chatfuel)

Depois de se carregar no botão *test this chatbot*, ambas as mensagens são apresentadas na conversa (Figura 26).

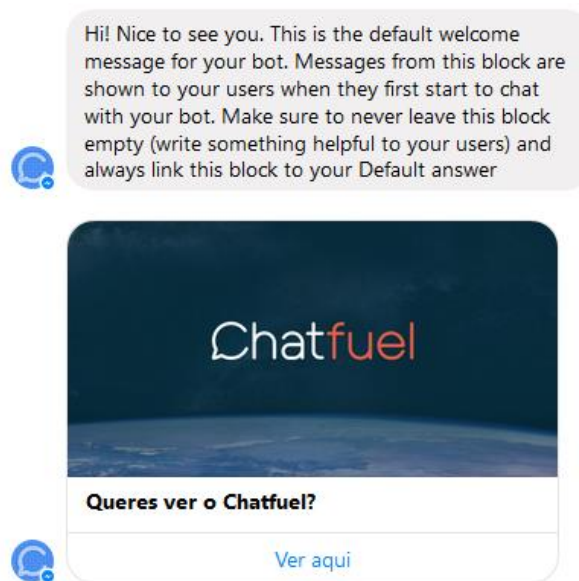


Figura 26 – Conversa criada no Chatfuel vistas no Facebook Messenger

Por fim, são apresentados alguns *plugins* que o Chatfuel tem (Figura 27). Um dos mais usados é o JSON API, que é muito útil pois permite gerar conteúdos dinâmicos.

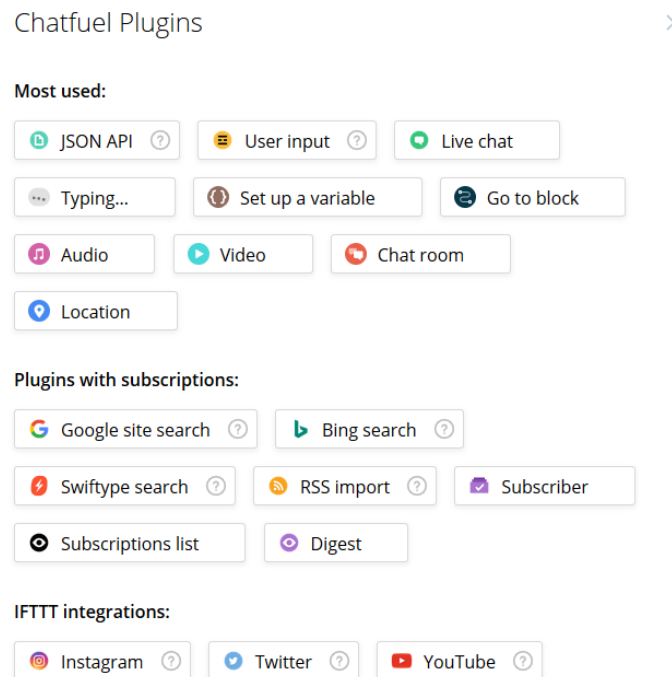


Figura 27 – *Plugins* associados ao Chatfuel

2.3.4. Síntese do Subcapítulo

Neste subcapítulo foram nomeadas e descritas algumas ferramentas que permitem a criação de sistemas de diálogo, mais direcionados para o Facebook Messenger e das mais utilizadas atualmente: Wit.ai, Api.ai e Chatfuel.

Na Tabela 9 e Tabela 10, é apresentada uma comparação entre as várias ferramentas, de modo a serem entendidas as diversas diferenças existentes.

Tabela 9 – Comparação entre as ferramentas de criação de *chatbots* (I)

| Termo a comparar | Wit.ai | Api.ai | Chatfuel |
|--|--------|--------|---------------|
| Ferramenta gratuita | Sim. | Sim. | Sim. |
| Associado o conceito de <i>intents</i> e <i>entities</i> | Sim. | Sim. | Não. |
| Domínios já predefinidos | Sim. | Sim. | Sim (poucos). |
| Criação do <i>chatbot</i> a partir do sítio <i>web</i> | Sim. | Sim. | Sim. |
| Associação do <i>chatbot</i> ao Facebook Messenger | Sim. | Sim. | Sim. |
| Integração com serviços | 1 | 14 | 2 |

Tabela 10 – Comparação entre as ferramentas de criação de *chatbots* (II)

| Termo a comparar | Wit.ai | Api.ai | Chatfuel |
|--|---------------|------------------------------------|-----------------|
| Necessárias competências de programação | Não. | Não. | Não. |
| Podem ser associadas competências de programação | Sim. | Sim. | Sim. |
| Tem alguma comunidade - fórum | Não. | Sim. | Sim. |
| Tem documentação | Sim. | Sim (é a ferramenta que tem mais). | Sim. |

Como é perceptível, as ferramentas são bastante semelhantes e apenas alguns detalhes as distinguem. O termo de comparação mais notório é o número de integrações com serviços disponíveis por cada ferramenta, na qual a Api.ai vence. Nesta dissertação o objetivo é apenas integrar com o Facebook Messenger, mas se no futuro for pretendido integrar com outro tipo de plataformas, é algo importante a ter em consideração.

Relativamente à documentação, apesar de todas fornecerem, também a ferramenta Api.ai tem um ponto a favor, porque é a que mais documentação variada dispõe, incluindo diversos exemplos de utilização. Para além disso também fornece um fórum onde é possível retirar dúvidas, ou até mesmo aprender com as dúvidas existentes que foram colocadas no mesmo.

Assim sendo, vendo como mais vantajosa a ferramenta Api.ai, não só pelos pontos diferenciais, mas também porque já foram realizados testes com todas as ferramentas, o protótipo a ser desenvolvido na fase de implementação, será desenvolvido recorrendo a esta mesma ferramenta. Este protótipo terá integração com o Facebook Messenger, mas como dispõe de outras treze interações, futuramente pode até ser interligado com outra plataforma.

3. Contexto e Descrição Técnica

Neste capítulo de contexto e descrição técnica, primeiramente será feito um enquadramento e apresentados detalhes da dissertação.

Para além disso, o mesmo tem como subcapítulo, a fase de análise e modelação, que reflete uma forte componente a nível de importância para o desenvolvimento de um projeto. Aqui será feita uma análise quanto ao público-alvo, incluindo dados estatísticos fornecidos pela Casa da Música, assim como os resultados de um questionário realizado neste âmbito. Serão também apresentados os requisitos do sistema, casos de uso referentes às funcionalidades do *chatbot*, um diagrama de atividades que permite de uma forma mais visual, entender o fluxo de interações do utilizador com o agente do *chatbot*, assim como um diagrama de arquitetura, de forma a representar a estrutura do sistema.

3.1. Enquadramento e Detalhes da Dissertação

A empresa Xarevision S.A. é considerada uma empresa líder em soluções de sistemas de informação para retalho. Mas atualmente está a investir na área de interfaces conversacionais, por realmente ser uma área que nos últimos anos tem vindo a crescer exponencialmente.

Assim sendo, surgiu a oportunidade de colaboração com a Casa da Música, uma instituição relacionada com eventos culturais, tendo em conta a sua agenda cultural. O objetivo é associar esta área, com o Facebook Messenger e criar um protótipo de um *chatbot* para obtenção de informações relativas à instituição, que é a Casa da Música.

Este protótipo tem como funcionalidade principal, a compra bilhetes para eventos que ocorrem na Casa da Música. Contudo tem também outras funcionalidades associadas: obter informações relativas a eventos, artistas, ou sobre a própria instituição.

Isto permite que os utilizadores consigam obter o que pretendem de forma mais simples e explicativa, ao invés de tentarem encontrar o mesmo, a partir do sítio *web* da Casa da Música. O público-alvo ao ter acesso ao *chatbot*, que está associado a uma página de Facebook, consegue comunicar com o agente do mesmo, através da própria linguagem natural, e solicitar o pretendido.

Este protótipo permitirá constatar quais as vantagens associadas ao uso de um *chatbot* para este fim e qual a importância que o público-alvo associa ao mesmo. Para além disso, permitirá obter uma visão de algo que ainda não existe e que de um ponto de vista funcional e de experiência de utilizador, tem bastante utilidade.

3.2. Análise e Modelação

Neste capítulo será apresentada uma análise relativa ao público-alvo relacionado com esta dissertação. Essa análise refere os *stakeholders* encontrados, assim como uma análise estatística relativa a dados fornecidos pela Casa da Música, e tendo em conta um questionário realizado a 292 pessoas.

Para além disso, também é aqui que feito o levantamento de todos os requisitos do sistema, casos de uso referentes às funcionalidades do *chatbot*, assim como um diagrama de atividades que reflete o fluxo de interações entre o utilizador e o agente do *chatbot*, de uma forma mais visual. Também um diagrama de arquitetura está presente neste subcapítulo, de modo a representar a estrutura do sistema.

3.2.1. Análise do Público-alvo

Numa fase inicial de análise foram identificados os seguintes *stakeholders* associados a esta dissertação: colaboradores da Casa da Música e os possíveis espectadores. O primeiro refere-se especificamente aos colaboradores da Casa da Música que usarão o *chatbot* para saberem como esse canal de comunicação funciona e ajudar os seus clientes nas atividades de suporte. Já o outro *stakeholder* refere-se a qualquer pessoa que tenha interesse em eventos culturais e que queira saber mais sobre a agenda cultural da Casa da Música, artistas ou sobre a própria instituição. Deste modo, podem ter interesse também em conhecer melhor a Casa da Música e fazerem visitas a esta instituição.

E para melhor compreender os interesses deste mesmo público-alvo, foi possível obter alguns dados estatísticos fornecidos pela Casa da Música, e referidos no subcapítulo 3.2.1.1. Para além disso, foi também realizado um questionário com algumas questões quantitativas, mas também qualitativas. A sua análise é apresentada no subcapítulo 3.2.1.2.

3.2.1.1. Dados Estatísticos

Recorrendo a dados estatísticos fornecidos pela Casa da Música, é possível fazer uma análise relativa ao público-alvo, de acordo com o que atualmente acontece.

Por exemplo, quanto à visita do sítio *web* da instituição através de utilizadores, tendo em conta o número de sessões, assim como a respetiva percentagem, a Tabela 11 apresenta esses valores para dez países.

Tabela 11 – Acesso ao sítio *web* da Casa da Música (dados estatísticos)

| País | Número de sessões | Percentagem de sessões (%) |
|----------------|-------------------|----------------------------|
| Portugal | 726 950 | 82,52 |
| Brasil | 24 820 | 2,82 |
| Espanha | 23 740 | 2,69 |
| Estados Unidos | 12 902 | 1,46 |
| França | 12 866 | 1,46 |
| Reino Unido | 12 252 | 1,39 |
| Alemanha | 11 547 | 1,31 |
| Países Baixos | 6 503 | 0,74 |
| Itália | 4 786 | 0,54 |
| Bélgica | 4 603 | 0,52 |

É possível verificar que Portugal é o país que realmente apresenta mais visitas ao sítio *web*, com uma percentagem de 82,52%, e o país com menos visitas, é a Bélgica, com uma percentagem de 0,52%. Contudo, há várias sessões vindas de países nos quais o idioma não é o português. Assim é importante que haja possibilidade de visualização do conteúdo em inglês, porque é o idioma mais utilizado mundialmente.

Já quanto a uma avaliação relativa à compra de bilhetes através da bilheteira da Casa da Música, comparativamente através do sítio *web*, a Figura 28 representa este estudo. Essa figura revela também uma percentagem referente à venda de bilhetes através dos agentes de venda (Worten) e juntamente, venda de bilhetes para eventos privados, mais direcionados aos trabalhadores da Casa da Música.

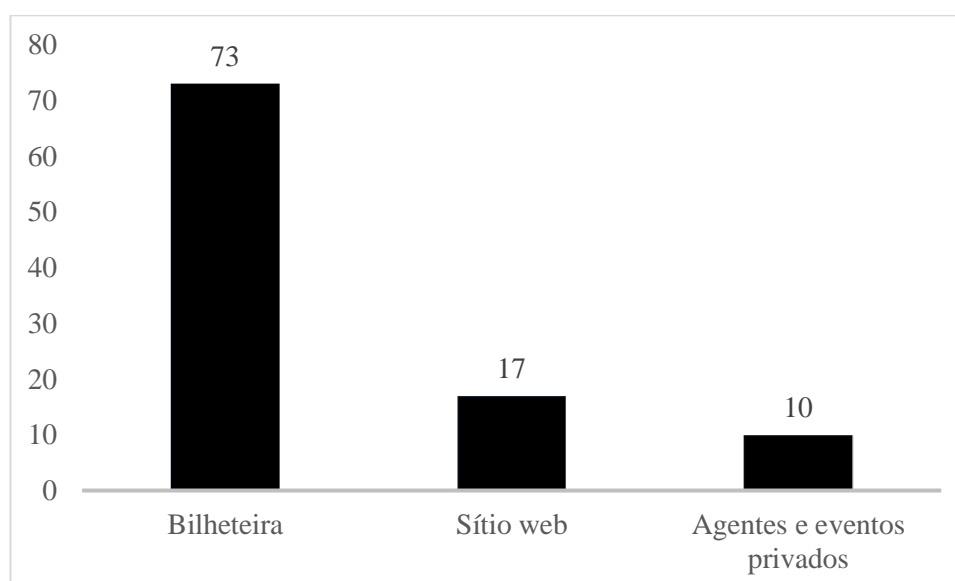


Figura 28 – Percentagem de compra de bilhetes (dados estatísticos)

É possível verificar que há um maior número percentual de pessoas que recorrem à compra de bilhetes através da bilheteira. Isto pode dever-se a dois fatores: o público não tem muitos conhecimentos relativamente a tecnologias ou então não sentem tanta facilidade nem segurança para efetuar a compra através do sítio *web*, porque não sentem tanto acompanhamento. Em loja podem sentir mais ajuda e apoio quanto a este processo. E é neste seguimento que o *chatbot* também tenta atuar, pois um dos seus objetivos é que os utilizadores consigam obter de forma fácil e eficaz, resultados aos seus pedidos, obtendo toda a ajuda necessária e um maior acompanhamento.

3.2.1.2. Questionário Inicial

De modo a se perceber e ser feita uma análise com uma componente mais estatística relativamente aos interesses dos utilizadores do *chatbot* a ser desenvolvido, foi realizado um questionário direcionado aos mesmos. Este questionário foi enviado a vários alunos de instituições da Universidade do Porto, trabalhadores das organizações envolvidas nesta dissertação, assim como a amigos.

Este consistiu em onze perguntas relativas à dissertação, e quatro de carácter pessoal. 292 pessoas responderam a este questionário e assim conseguiu-se fazer um levantamento e análise dos dados referentes. Este conjunto de pessoas, têm idades compreendidas entre dois grupos: menores de 18 e entre os 56 aos 65. Maioritariamente, com 76%, as respostas foram facultadas por elementos inseridos no grupo de idades entre os 18 e os 25, e do sexo feminino (53%).

Várias vertentes profissionais foram também encontradas neste estudo, que permitiram uma maior diversidade de opiniões relativas ao estudo em questão, tais como: estudantes, *designers*, engenheiros, biólogos, docentes, consultores, músicos, tradutores, programadores, físicos, entre

outros. Do mesmo modo, também foram obtidas respostas de várias nacionalidades: portuguesa, brasileira, britânica, iraniana, eslovaca e checa, sendo que houve uma maior componente de pessoas portuguesas.

Foram realizadas maioritariamente questões com uma componente mais quantitativa e dentro desta, seis questões referem uma escala entre um (baixo/nada útil) a cinco (alto/muito útil). Para as mesmas foi feita uma análise através do *software* IBM SPSS Statistics, de modo a se poder observar a média, desvio-padrão, e a frequência. As questões são as seguintes, com as respetivas escalas (Tabela 12):

Tabela 12 – Questões de componente quantitativa - questionário inicial

| ID | Questão | Escala |
|-----|---|--------------------------------|
| Q01 | Qual o seu nível de interesse em eventos da Casa da Música? | Baixo – Alto (1 – 5) |
| Q02 | Qual o seu nível de conhecimento relativo a <i>chatbots</i> ? | Baixo – Alto (1 – 5) |
| Q03 | Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse comprar bilhetes para eventos? | Nada útil – Muito útil (1 – 5) |
| Q04 | Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse obter informações sobre eventos? | Nada útil – Muito útil (1 – 5) |
| Q05 | Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse obter informações sobre artistas? | Nada útil – Muito útil (1 – 5) |
| Q06 | Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse obter informações sobre a instituição (Casa da Música)? | Nada útil – Muito útil (1 – 5) |

Nos Anexos (Anexo 1 – Tabelas de Frequência) são apresentadas as tabelas de frequência referentes a cada uma destas respostas, para uma análise mais pormenorizada. Estas referem as frequências de resposta, tendo em conta cada hipótese disponível quanto à escala apresentada, assim como percentagens.

Já a Tabela 13, é uma tabela descritiva que apresenta para cada uma das questões, o número de respostas obtidas, o valor mínimo e máximo de resposta, a média das mesmas, assim como o desvio padrão, representando a dispersão das respostas. Esta tabela foi gerada através do *software* IBM SPSS Statistics.

Tabela 13 - Tabela descritiva (Q01 – Q06) - questionário inicial

| ID | N | Mínimo | Máximo | Média | Desvio Padrão |
|-----|-----|--------|--------|-------|---------------|
| Q01 | 292 | 1 | 5 | 3,1 | 1,2 |
| Q02 | 292 | 1 | 5 | 2,4 | 1,3 |
| Q03 | 292 | 1 | 5 | 3,9 | 1,1 |
| Q04 | 292 | 1 | 5 | 3,6 | 1,0 |
| Q05 | 292 | 1 | 5 | 3,5 | 1,0 |
| Q06 | 292 | 1 | 5 | 3,3 | 1,0 |

De uma forma mais visual, o gráfico representado através da Figura 29, apresenta para cada questão (Q01 a Q06), a média de resposta, assim como o desvio padrão relativo a cada uma.

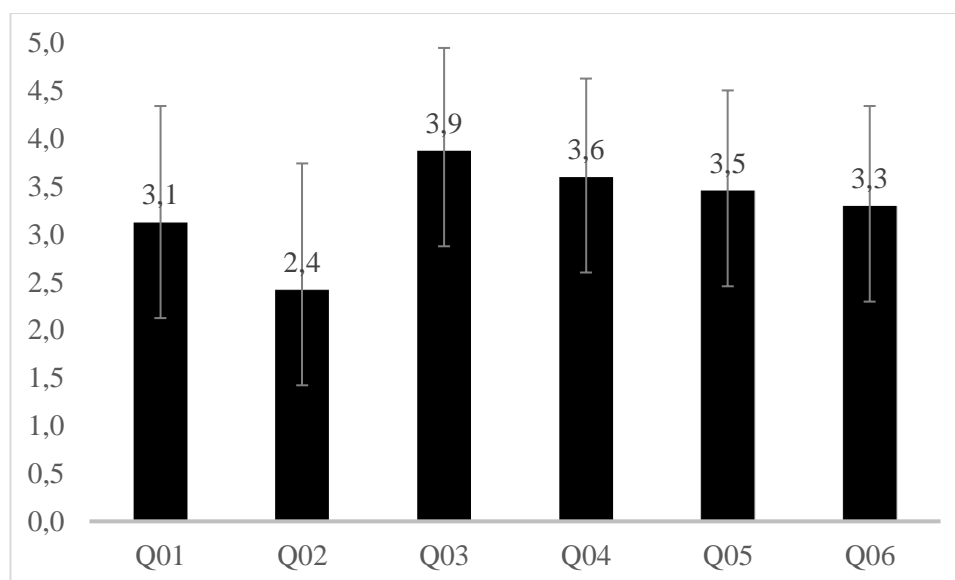


Figura 29 - Média de respostas (Q01 – Q06) - questionário inicial

Ainda bastantes pessoas referem que não têm grande conhecimento relativo a *chatbots* (Q02), mas muitas delas acharam a ideia desta dissertação, algo que pode ter grande valor. A funcionalidade que para o público teria mais interesse, é realmente a de compra de bilhetes para eventos da Casa da Música, através do Facebook Messenger (Q03), pois é a que apresenta uma maior média. No geral, as perguntas referentes às funcionalidades a serem desenvolvidas no protótipo (Q03 a Q06), apresentam uma média superior a 3, o que significa que maioritariamente, a utilidade está mais próxima do máximo, do que do mínimo.

Outras três questões evidenciam a frequência com que as pessoas que responderam ao inquérito, assistem a eventos na Casa da Música, e até mesmo a frequência relativa ao uso do Facebook e Facebook Messenger (Tabela 14).

Tabela 14 – Questões relativas a frequências - questionário inicial

| ID | Questão | Possíveis Respostas |
|-----|---|---|
| Q07 | Com que frequência assiste a eventos na Casa da Música? | Nunca Uma vez por ano Uma vez por mês Mais frequentemente |
| Q08 | Costuma usar o Facebook com que frequência? | Nunca Uma vez por ano Uma vez por mês Uma vez por semana Uma vez por dia Várias vezes ao dia |
| Q09 | Costuma usar o Facebook Messenger com que frequência? | Nunca Uma vez por ano Uma vez por mês Uma vez por semana Uma vez por dia Várias vezes ao dia |

Para cada uma destas, é apresentado um gráfico que reflete a percentagem referente ao número de respostas consoante as hipóteses associadas a cada pergunta - Figura 30, Figura 31 e Figura 32.

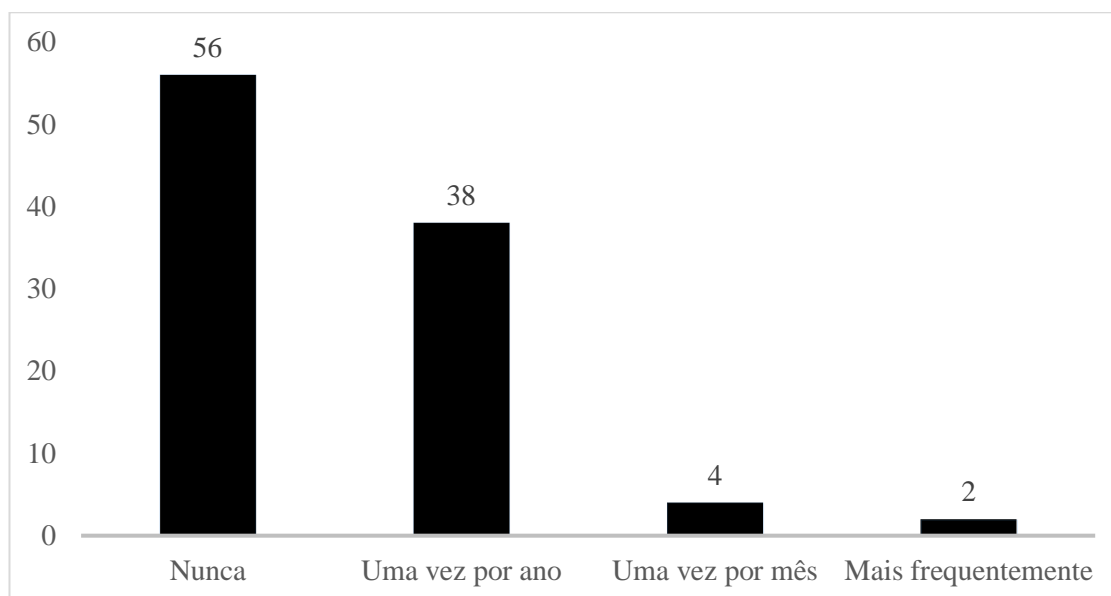


Figura 30 – Percentagem de respostas à questão Q07 - questionário inicial

Tendo em conta o público que respondeu a este questionário, percebe-se que são mais as pessoas que não assistem ou que assistem poucas vezes a eventos na Casa da Música, do que as que os procuram ver mais frequentemente.

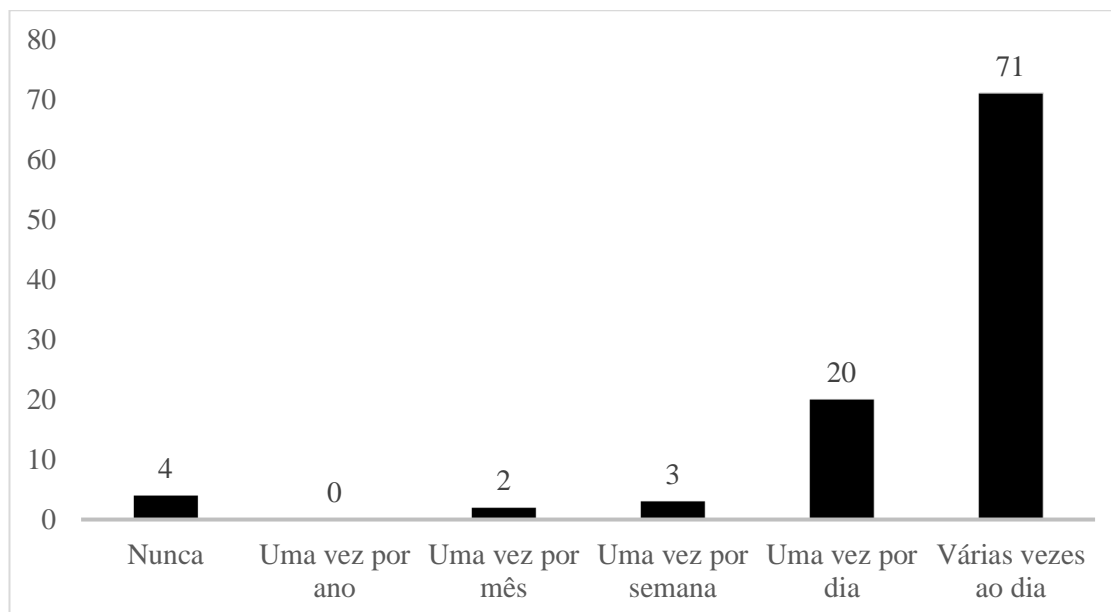


Figura 31 – Percentagem de respostas à questão Q08 - questionário inicial

Já quanto à frequência de uso do Facebook, o caso já apresenta algumas diferenças. Realmente a maioria utiliza esta rede social várias vezes ao dia, ou pelo menos uma vez por dia. Há também algo que se pode constatar: as pessoas não usam de todo o Facebook ao invés de utilizarem uma vez por ano.

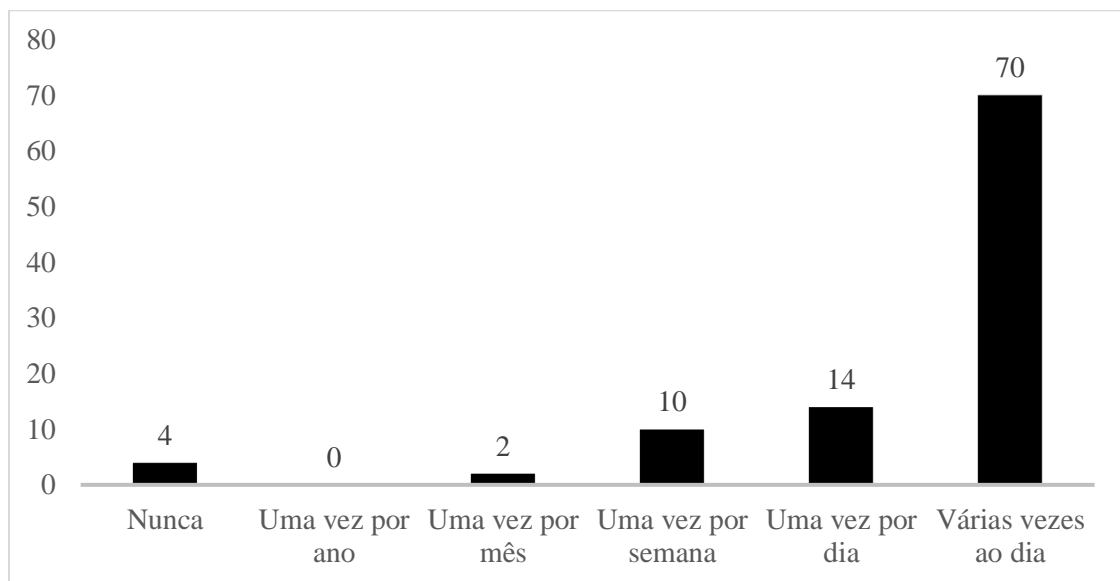


Figura 32 - Percentagem de respostas à questão Q09 - questionário inicial

Do mesmo modo se verifica que a frequência de uso do Facebook Messenger, apresenta praticamente o mesmo tipo de percentagens que a questão referente à frequência de uso do Facebook. Aqui constata-se que há um maior número de pessoas que respondeu uma vez por semana, passando de 3% para 10%, comparando a questão Q08 para a Q09.

E a última questão, foi colocada para perceber que outro tipo de funcionalidades as pessoas achariam úteis, para além das referidas neste questionário. Na Tabela 15 e Tabela 16, estão listadas essas mesmas funcionalidades.

Tabela 15 – Funcionalidades sugeridas (I)

| Funcionalidade |
|---|
| 1. Consultar eventos passados/futuros. |
| 2. Verificação de lugares e comentários organizados por lugar para se ter <i>feedback</i> do mesmo. |
| 3. Notificações relativas a eventos subscritos. |
| 4. Informações atualizadas sobre os artistas ou o espetáculo em si, seria realmente bom. |
| 5. Avisos sobre eventos: cancelamentos, atrasos, lembretes para não chegar atrasado, etc. |
| 6. Algo que seria interessante, era conseguir sugerir eventos consoante as escolhas do utilizador. |
| 7. Promover os eventos de data mais próxima. |
| 8. Visita virtual da instituição. |
| 9. Obter a localização do utilizador e sugerir horários de transportes públicos (metro e/ou autocarro). |
| 10. Datas, <i>dresscode</i> , eventos futuros, preços, descontos, horários, entre outros. |

Tabela 16 – Funcionalidades sugeridas (II)

| Funcionalidade |
|--|
| 11. Dizer informações do trânsito e agenda. |
| 12. Consultar horários e futuros eventos. |
| 13. Relativamente às informações sobre a instituição, deveria ter também um sistema para dar as coordenadas do local, e dentro deste, o local mais próximo por onde a pessoa deve entrar para chegar ao seu lugar. |
| 14. Passatempos. |
| 15. Outros locais além da Casa da Música. |

No geral, o parecer dos participantes quanto a esta questão foi muito positivo, porque mesmo não sendo uma pergunta de resposta obrigatória, houve propostas de funcionalidades que o público vê como úteis a integrar neste *chatbot*. Isto significa que, mesmo não sendo um grande número de pessoas, houve um certo interesse por parte de quem respondeu a este questionário.

Muitas das sugestões incidiram nos eventos disponibilizados pela Casa da Música, mais concretamente na possibilidade de existência de vários tipos de notificação.

3.2.2. Requisitos

Tendo em conta o problema a solucionar, foram definidos os requisitos do sistema associados a esta dissertação, para o desenvolvimento de um protótipo para a Casa da Música (Tabela 17 e Tabela 18).

Tabela 17 – Lista de requisitos (I)

| ID | Descrição |
|-------|---|
| REQ-1 | Deve ser selecionada uma ferramenta que permita a criação de <i>chatbots</i> para o Facebook Messenger. |
| REQ-2 | Criar um <i>chatbot</i> com recurso à ferramenta selecionada. |
| REQ-3 | Criar um agente conversacional que tenha conhecimentos relativos à área de negócio da Casa da Música, incluindo eventos, artistas e bilhetes. |
| REQ-4 | O utilizador deve conseguir ter acesso ao <i>chatbot</i> através do Facebook Messenger (devido a uma página de Facebook existente). |
| REQ-5 | O utilizador deve conseguir comunicar com o agente do <i>chatbot</i> através de linguagem natural. |
| REQ-6 | O utilizador deve conseguir obter informações relativas à instituição, que é a Casa da Música. |

Tabela 18 – Lista de requisitos (II)

| ID | Descrição |
|-------|--|
| REQ-7 | O utilizador deve conseguir obter informações relativas a artistas, especificando o nome do artista. |
| REQ-8 | O utilizador deve conseguir obter informações relativas a eventos que ocorrem na Casa da Música, indicando o nome do evento. |
| REQ-9 | O utilizador deve conseguir comprar bilhetes através do <i>chatbot</i> , tendo em conta um evento em particular. |

Esses mesmos requisitos têm início na seleção de uma ferramenta para criação de um *chatbot* com integração para o Facebook Messenger, até às funcionalidades inerentes ao longo do desenvolvimento deste protótipo.

3.2.3. Casos de Uso

Relativamente à Engenharia de Software, o modelo de casos de uso é um modelo das funções pretendidas que serve como contrato entre o cliente e o responsável pelo desenvolvimento. Permite assim indicar quais os requisitos que a solução a desenvolver deve cumprir, segundo o ponto de vista dos atores.

Os *stakeholders* encontrados são os trabalhadores da Casa da Música e os espectadores, mas será considerado um único ator para o levantamento dos casos de uso, que é o utilizador. Ambos os *stakeholders* conseguem realizar as mesmas tarefas, e assim sendo, será considerado apenas este ator.

Seguidamente não só será apresentado o diagrama UML de casos de uso (Figura 33), como também uma explanação de cada caso de uso em específico (Tabela 19 a Tabela 30).

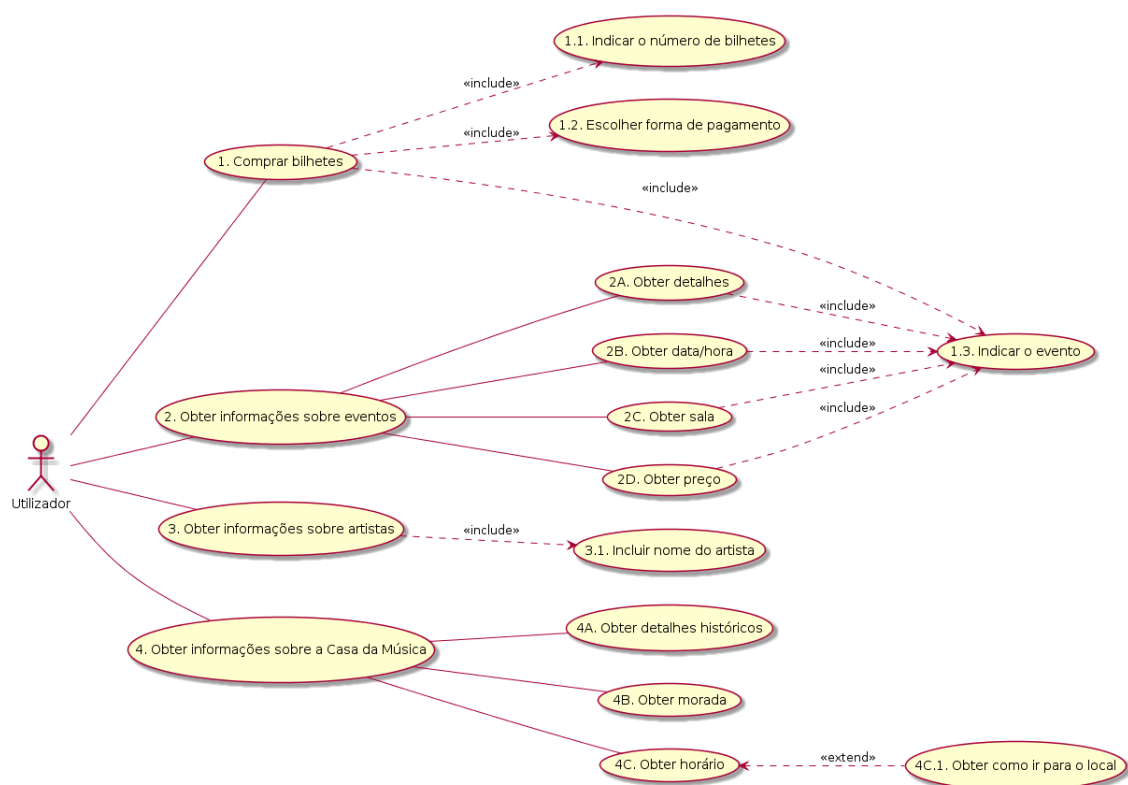


Figura 33 – Diagrama de casos de uso

Em primeiro lugar, o utilizador deve conseguir efetuar o pedido de compra de bilhetes associados a eventos, através do uso de linguagem natural (Tabela 19).

Tabela 19 – Descrição do caso de uso 1

| ID | 1 |
|---------------|---|
| Nome | Comprar bilhetes. |
| Descrição | Através de linguagem natural, o utilizador consegue comprar bilhetes para um evento. |
| Pré condições | - |
| Pós condições | O utilizador tem de incluir qual o evento, o tipo e número de bilhetes e escolher a forma de pagamento. |

Deve ser possível a indicação do número de bilhetes que o utilizador pretende, durante o caso de uso de compra de bilhetes (Tabela 20).

Tabela 20 - Descrição do caso de uso 1.1

| ID | 1.1 |
|---------------|---|
| Nome | Indicar o número de bilhetes. |
| Descrição | Tendo como objetivo comprar bilhetes para eventos, o utilizador deve indicar qual o número de bilhetes que pretende adquirir. |
| Pré condições | Ter iniciado o processo de compra de bilhetes. |
| Pós condições | - |

Dentro desta componente, o utilizador tem de indicar qual a forma de pagamento que pretende utilizar para a compra dos seus bilhetes (Tabela 21). O processo de compra não faz parte do âmbito desta dissertação, ou seja, não é suposto comprar verdadeiramente os bilhetes, mas sim simular este processo. Assim sendo está incluído nos casos de uso, a escolha da forma de pagamento pelo utilizador, como fase final.

Tabela 21 – Descrição do caso de uso 1.2

| ID | 1.2 |
|---------------|--|
| Nome | Escolher forma de pagamento. |
| Descrição | Tendo como objetivo comprar bilhetes para eventos, o utilizador tem de indicar qual a forma de pagamento que pretende utilizar para a compra dos mesmos. |
| Pré condições | Ter iniciado o processo de compra de bilhetes. |
| Pós condições | - |

Para a compra de bilhetes, o utilizador tem de associar um evento, e por isso trata-se de um caso de uso de inclusão do anterior (casos de uso 1). Como um utilizador deve conseguir proceder à obtenção de informação relativa a eventos, este caso de uso é também uma inclusão do caso de uso 2, tendo em conta todos os seus casos de uso envolventes (2A, 2B, 2C e 2D) (Tabela 22).

Tabela 22 – Descrição do caso de uso 1.3

| ID | 1.3 |
|---------------|---|
| Nome | Indicar o evento. |
| Descrição | Tendo como objetivo obter informações sobre eventos, ou comprar bilhetes, o utilizador deve incluir qual o espetáculo em questão. |
| Pré condições | Ter iniciado o processo de obtenção de informações relativas a eventos ou de compra de bilhetes. |
| Pós condições | - |

O utilizador deve conseguir também obter informações sobre eventos que ocorrem na Casa da Música, do mesmo modo que o caso de uso anterior (Tabela 23).

Tabela 23 – Descrição do caso de uso 2

| ID | 2 |
|---------------|--|
| Nome | Obter informações sobre eventos. |
| Descrição | Através de linguagem natural, o utilizador consegue obter informações sobre eventos que fazem parte da agenda cultural da Casa da Música, tendo em conta quatro subtipos: detalhes, data/hora, sala e preço dos bilhetes para esse evento. |
| Pré condições | - |
| Pós condições | - |

Ao pedir informação sobre eventos, o utilizador pode ter em consideração, quatro tipos de pedido. O primeiro é apresentado na Tabela 24 e diz respeito à obtenção de detalhes relativos a um evento indicado pelo utilizador.

Tabela 24 – Descrição do caso de uso 2A

| ID | 2A |
|---------------|---|
| Nome | Obter detalhes. |
| Descrição | Este é um dos subtipos de pedido relativo à informação sobre eventos. O utilizador consegue obter detalhes sobre um evento. |
| Pré condições | - |
| Pós condições | O utilizador deve incluir o nome de um evento. |

O segundo refere-se à data/hora de um evento (Tabela 25). Ao fazer esse pedido, o utilizador consegue obter essa informação.

Tabela 25 – Descrição do caso de uso 2B

| ID | 2B |
|---------------|---|
| Nome | Obter data/hora. |
| Descrição | Este é um dos subtipos de pedido relativo à informação sobre eventos. O utilizador consegue obter a data/hora de um evento. |
| Pré condições | - |
| Pós condições | O utilizador deve incluir o nome de um evento. |

Com o terceiro, o utilizador é capaz de obter qual a sala da Casa da Música onde vai ocorrer um determinado evento (Tabela 26).

Tabela 26 – Descrição do caso de uso 2C

| ID | 2C |
|---------------|--|
| Nome | Obter sala. |
| Descrição | Este é um dos subtipos de pedido relativo à informação sobre eventos. O utilizador consegue obter a sala onde vai ocorrer um evento. |
| Pré condições | - |
| Pós condições | O utilizador deve incluir o nome de um evento. |

Por último, o quarto subtipo, permite ao utilizador obter o preço dos bilhetes que dizem respeito a um evento (Tabela 27).

Tabela 27 – Descrição do caso de uso 2D

| ID | 2D |
|---------------|--|
| Nome | Obter preço. |
| Descrição | Este é um dos subtipos de pedido relativo à informação sobre eventos. O utilizador consegue obter o preço dos bilhetes de um determinado evento. |
| Pré condições | - |
| Pós condições | O utilizador deve incluir o nome de um evento. |

Outro caso de uso é o de obtenção de informações sobre artistas. Em comum com os casos de uso anteriores, para tal, é utilizada linguagem natural (Tabela 28).

Tabela 28 – Descrição do caso de uso 3

| ID | 3 |
|---------------|---|
| Nome | Obter informações sobre artistas. |
| Descrição | Através de linguagem natural, o utilizador consegue obter informações sobre artistas. |
| Pré condições | - |
| Pós condições | O utilizador pode incluir uma área e/ou nome do artista. |

Tendo em conta o caso de uso anterior, o utilizador deve conseguir especificar o nome do artista que pretende obter mais informações (Tabela 29).

Tabela 29 – Descrição do caso de uso 3.1

| ID | 3.1 |
|---------------|---|
| Nome | Incluir nome do artista. |
| Descrição | Tendo como objetivo obter informações sobre artistas, o utilizador pode incluir o nome do artista que pretende saber mais detalhes. |
| Pré condições | Ter iniciado o processo de obtenção de informações relativas a artistas. |
| Pós condições | - |

O utilizador deve conseguir obter informações sobre a Casa da Música, conversando através do uso de linguagem natural, com o agente do *chatbot* (Tabela 30).

Tabela 30 – Descrição do caso de uso 4

| ID | 4 |
|---------------|--|
| Nome | Obter informações sobre a Casa da Música. |
| Descrição | Através de linguagem natural, o utilizador consegue obter informações sobre a instituição, conversando com o agente do <i>chatbot</i> . Este abrange três subtipos de casos de uso: detalhes históricos, morada e horário. |
| Pré condições | - |
| Pós condições | - |

O utilizador pode obter informações sobre a Casa da Música, tendo em conta uma vertente mais histórica (Tabela 31).

Tabela 31 – Descrição do caso de uso 4A

| ID | 4A |
|---------------|---|
| Nome | Obter detalhes históricos. |
| Descrição | Este é um dos subtipos de pedido relativo à informação sobre a instituição. O utilizador consegue obter detalhes históricos sobre a Casa da Música. |
| Pré condições | - |
| Pós condições | - |

Outra das vertentes refere-se à obtenção da morada da instituição (Tabela 32). Aqui existe uma pós condição opcional, por isso é que se trata de uma extensão.

Tabela 32 – Descrição do caso de uso 4B

| ID | 4B |
|---------------|---|
| Nome | Obter morada. |
| Descrição | Este é um dos subtipos de pedido relativo à informação sobre a instituição. O utilizador consegue obter detalhes históricos sobre a Casa da Música. |
| Pré condições | - |
| Pós condições | O utilizador pode obter como ir para o local. |

Se o utilizador pretender saber como ir para o local, após indicada a morada, pode também obter essa informação (Tabela 33).

Tabela 33 – Descrição do caso de uso 4B.1

| ID | 4B.1 |
|---------------|---|
| Nome | Obter como ir para o local. |
| Descrição | Este caso de uso é uma extensão do 4B. Após indicada a morada, o utilizador tem a possibilidade de saber como ir para o local através de autocarro, metro ou carro. |
| Pré condições | - |
| Pós condições | - |

E a última vertente relaciona-se com a opção de o utilizador conseguir obter o horário da Casa da Música (Tabela 34).

Tabela 34 – Descrição do caso de uso 4C

| ID | 4C |
|---------------|--|
| Nome | Obter horário. |
| Descrição | Este é um dos subtipos de pedido relativo à informação sobre a instituição. O utilizador consegue obter o horário da Casa da Música. |
| Pré condições | - |
| Pós condições | - |

3.2.4. Diagrama de Atividades

Para melhor entendimento do processo de interação a ser realizado, foi elaborado um diagrama de atividades que representa a síntese do fluxo de interações entre o utilizador e o agente do *chatbot* – Figura 34.

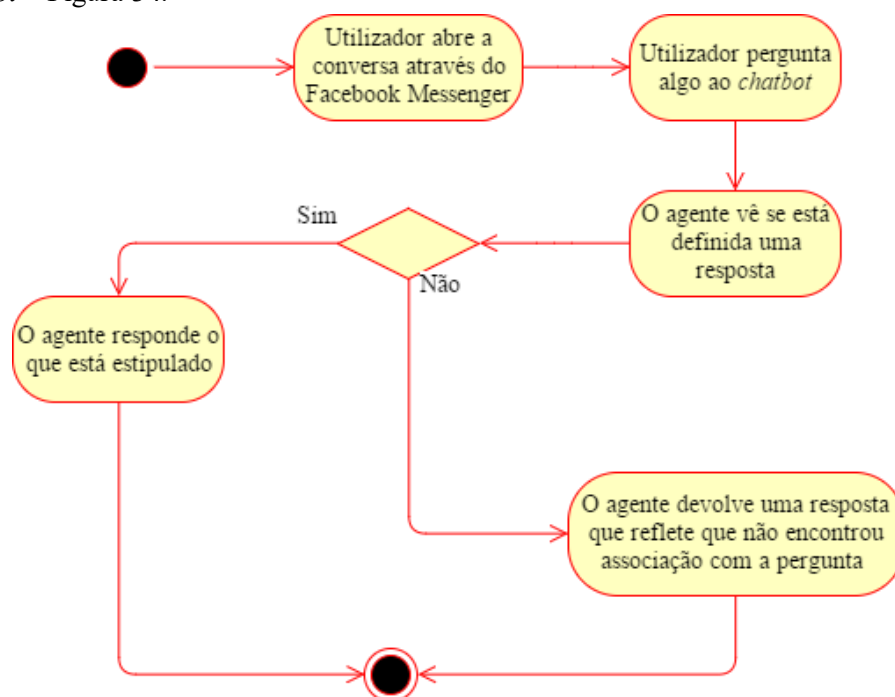


Figura 34 – Diagrama de atividades do fluxo de interações do utilizador/agente

Em primeiro lugar o utilizador necessita de iniciar uma conversa com o agente do *chatbot*, através do Facebook Messenger. Após essa fase, o utilizador faz o pedido que pretende, e o agente processa a informação, vendo se existe alguma resposta de acordo com aquilo que foi requerido. Em caso positivo, é devolvida uma resposta referente a esse pedido, mas em caso contrário, o agente devolve uma resposta, indicando que o que o utilizador pediu não foi encontrado. De uma forma menos técnica, a Figura 35, representa então esse fluxo.

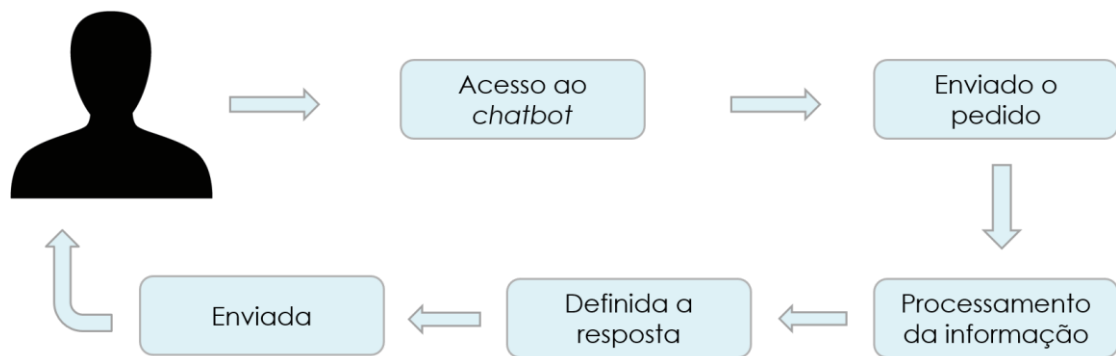


Figura 35 – Fluxo de interações do utilizador com o *chatbot*

Começa pelo utilizador, que tem acesso ao *chatbot* através de uma página de Facebook associada. Este envia o seu pedido através da própria linguagem natural e essa informação é processada para dar origem a uma resposta. Essa resposta é então definida e enviada para o utilizador.

3.2.5. Diagrama de Arquitetura

De modo a entender e clarificar a estrutura do sistema, relacionando os seus requisitos, o diagrama de arquitetura é um artefacto muito útil. Tendo isto em consideração, foi desenvolvido um diagrama de arquitetura que reflete as ligações entre todas as plataformas e ferramentas (Figura 36), começando pelo utilizador.

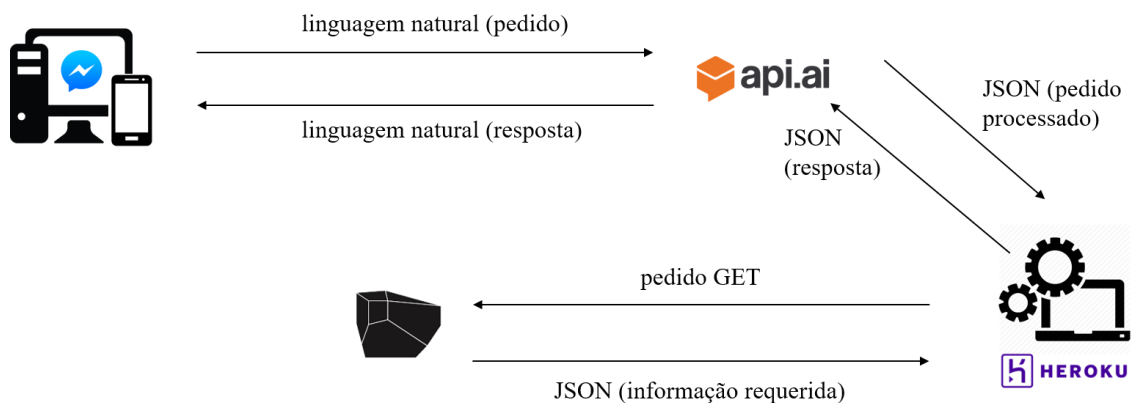


Figura 36 – Diagrama de arquitetura

O utilizador inicia uma conversa através do Facebook Messenger, utilizando linguagem natural. Esse pedido é enviado para a ferramenta Api.ai, podendo haver comunicação com o *back-end*, caso o pedido necessite de uma resposta dinâmica (relativa a artistas e eventos). Em caso afirmativo, comunica com o *back-end* que está alojado no Heroku, e é nesta fase que é feito um

ou mais pedidos GET¹⁰ ao *web service* da Casa da Música, consoante aquilo que for pretendido. Sempre que há comunicação com a *Api.ai*, um JSON está envolvido no processo. Para o utilizador obter então uma resposta, o procedimento inverte a ordem.

3.3. Síntese do Capítulo

Este capítulo iniciou-se com um enquadramento à dissertação, incluindo todos os detalhes associados.

Seguidamente, foi descrita a fase de análise e modelação inicial para a definição do protótipo funcional do *chatbot*. Dentro desta fase, foi realizado um estudo ao possível público-alvo, tendo em conta duas vertentes: dados estatísticos fornecidos pela Casa da Música e um questionário inicial, realizado a 292 pessoas com diferentes nacionalidades. Estes dados permitiram perceber qual o público atual da instituição, assim como qual o interesse dos possíveis utilizadores quanto ao sistema de diálogo textual a ser desenvolvido. Após este estudo foram definidos os requisitos e os casos de uso associados à dissertação, na componente de desenvolvimento do protótipo. Foi também criado um diagrama de atividades que define o fluxo de interações entre o utilizador e o sistema, assim como um diagrama de arquitetura para entender quais as ligações entre as plataformas e ferramentas no âmbito do desenvolvimento. Todos estes tópicos foram realizados e abordados em subcapítulos referentes.

Todos estes passos tratados nos vários subcapítulos, permitiram uma melhor definição do protótipo a ser desenvolvido, através das fases de análise e *design*.

¹⁰ Método usado para solicitação de uma resposta entre um cliente e um servidor.

4. Implementação

Neste capítulo de Implementação, serão tidos em conta todos os processos e fases efetuadas ao longo do desenvolvimento de todas as funcionalidades propostas: compra de bilhetes para eventos da Casa da Música, devolução de informações relativas a eventos, artistas e instituição. Para chegar a estas funcionalidades base, foram necessários alguns recursos, e também estes serão abordados neste capítulo.

E uma vez que segundo o estudo realizado durante o levantamento do estado da arte, se optou pela ferramenta Api.ai para criação do protótipo do *chatbot*, o primeiro capítulo aborda esta ferramenta. Encontram-se descritos todos os passos necessários para a criação do agente que comunica com o utilizador, assim como para criação de diálogo. Também é exposta a forma como foi feita a integração com o Facebook Messenger, e como foi realizada a ligação com o *back-end* que foi criado em PHP.

Já no capítulo seguinte do *Back-end*, é referida a fase de criação do mesmo, tendo em conta o seu propósito, a ligação com o Heroku, assim como com um repositório criado no Bitbucket.

Os capítulos seguintes referem em que consistiu a implementação das quatro funcionalidades base, referidas acima.

4.1. Api.ai

Neste subcapítulo serão abordadas todas as fases necessárias para a criação e definição do *chatbot* na Api.ai, desde a criação do agente, de entidades, integração com o Facebook Messenger, criação de *intents*, até à ligação com o servidor.

4.1.1. Criação do Agente

O primeiro passo a ter em consideração na ferramenta Api.ai é a criação de um agente. Neste caso foi criado o agente cujo nome é CM_Sara.

Nesta fase é necessário definir o idioma e após ser escolhido, não é possível alterar. Nos dados estatísticos (subcapítulo 3.2.1.1) e no questionário inicial (subcapítulo 3.2.1.2), constatou-se que o público não é apenas de nacionalidade portuguesa. Contudo, de modo a tentar mudar a realidade de uma maior percentagem de pessoas portuguesas, associadas às visitas no sítio *web* da Casa da Música e como o inglês é o idioma universal, foi selecionada a língua inglesa. Para além disso, a Api.ai apresenta um melhor funcionamento a nível de *machine learning*, e suporta muitas mais funcionalidades, quando o idioma é o inglês. Foi possível concluir isto porque foram realizados alguns testes antes da implementação. Em inglês, algo que é fácil obter resposta, em português não se conseguia, por exemplo. Por outro lado, é também possível consultar o anexo (8.2) que reflete os domínios existentes para cada idioma e nota-se que realmente o inglês é o mais completo. Para além disso, talvez se for criado um *chatbot*, atraísse mais estrangeiros.

Também o tipo de agente é necessário definir: público ou privado. Para haver mais confidencialidade, optou-se pelo tipo privado.

Após se ter criado o agente, são geradas imediatamente duas chaves (acesso de cliente e desenvolvedor), que permitirão depois a comunicação com o *backend* (capítulo 4.2).

Na Figura 37 é possível visualizar a página de definições do agente criado, onde se pode encontrar o tipo, descrição, idioma, fuso horário padrão, assim como as chaves referidas.


CM_Sara

SAVE

General

ML Settings

Export and Import



AGENT TYPE

☐ Public ☒ Private

DESCRIPTION

Describe your agent...

LANGUAGE ?

English

DEFAULT TIME ZONE

(GMT-5:00) America/New_Y...

API keys




| | | |
|------------------------|----------------------------------|---|
| Client access token | e7e3f92e402b43a7964aa269e3db22e8 |   |
| Developer access token | fe7b2ce781cf49e0b502c4975362230d |  |

Figura 37 – Página de definições do agente criado

4.1.2. Criação de Entidades

A segunda fase deste processo consiste na criação de entidades que permitem ao agente identificar determinadas palavras, inseridas pelo utilizador.

Por exemplo, uma entidade pode ser a *@artist*, que reflete nomes de artistas que os utilizadores podem pedir mais informações (Figura 38).

| Artist Name | Synonyms |
|------------------|--|
| Gileno Santana | Gileno Santana |
| Wolfgang Mozart | Wolfgang Mozart, Mozart, Wolfgang Amadeus Mozart |
| Sara Serpa | Sara Serpa |
| Sara Barros | Sara Barros |
| João Lencastre | João Lencastre |
| Thomas Zehetmair | Thomas Zehetmair |
| Paulo Barros | Paulo Barros |

Figura 38 – Entidade @artist

Para cada artista podem também ser definidos sinónimos. Por exemplo, se o utilizador disser “Mozart”, a Api.ai sabe que o nome correto é Wolfgang Mozart.

4.1.3. Integração com o Facebook Messenger

Uma vez que um dos objetivos desta dissertação é o uso do protótipo através do Facebook Messenger, foi necessário tratar da integração do agente criado, com esta plataforma.

Assim sendo, foi criada uma página de Facebook (CM_Sara) e uma aplicação na página de desenvolvedores do Facebook, para então associar o produto Messenger à página criada. Só assim é que se torna possível a ligação entre a Api.ai e o Facebook Messenger.

No produto Messenger, desta aplicação criada, foi então criado um *webhook*, subscrevendo a página de Facebook criada anteriormente (Figura 39).

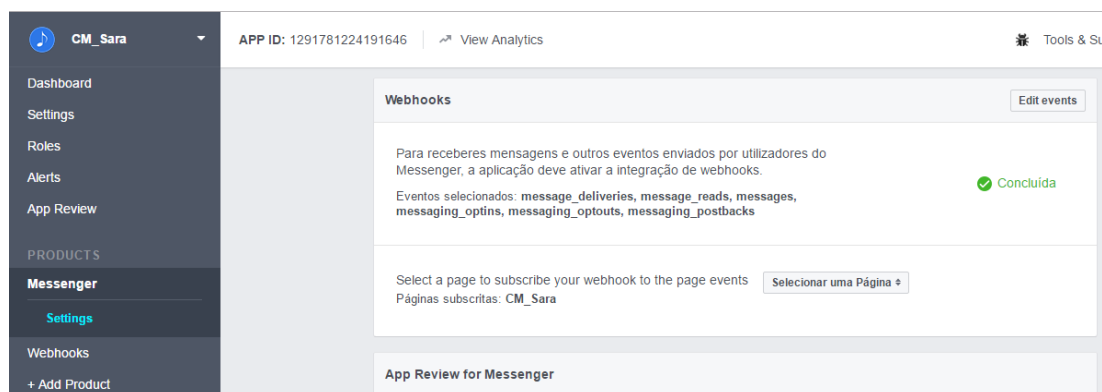
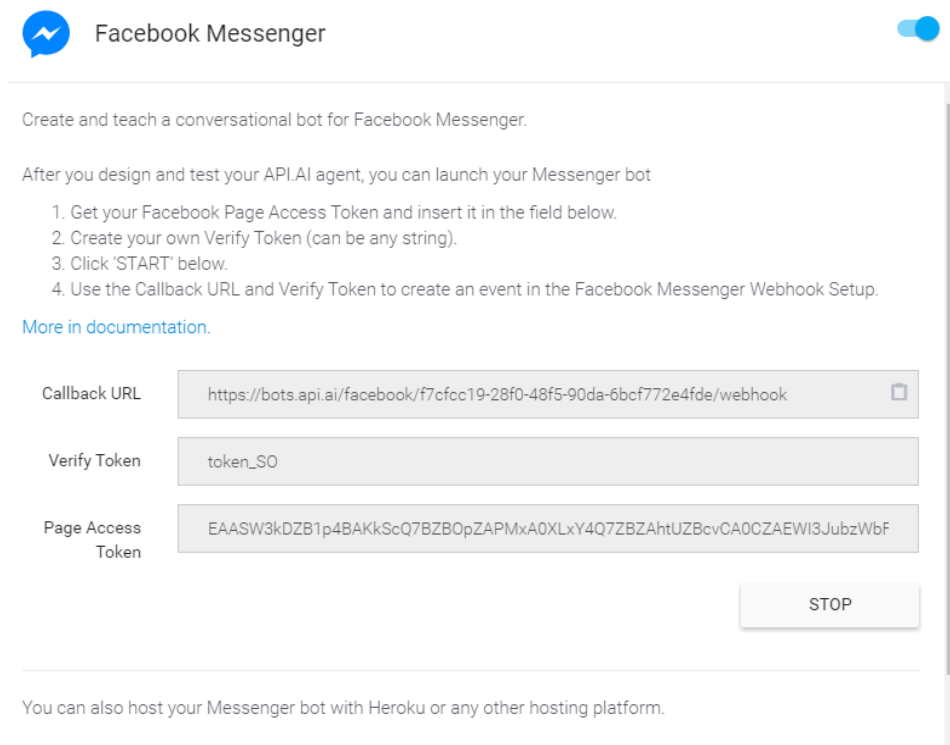


Figura 39 – Webhooks associados à aplicação CM_Sara

A fase seguinte consistiu na definição da integração pretendida, do lado da Api.ai (Figura 40).



Facebook Messenger

Create and teach a conversational bot for Facebook Messenger.

After you design and test your API.AI agent, you can launch your Messenger bot

1. Get your Facebook Page Access Token and insert it in the field below.
2. Create your own Verify Token (can be any string).
3. Click 'START' below.
4. Use the Callback URL and Verify Token to create an event in the Facebook Messenger Webhook Setup.

[More in documentation.](#)

Callback URL:

Verify Token:

Page Access Token:

You can also host your Messenger bot with Heroku or any other hosting platform.

Figura 40 – Integração com o Facebook Messenger (Api.ai)

Para tal foi necessário definir um *token* de verificação, assim como o *token* de acesso à página pretendida. Este mesmo *token* pode ser encontrado, também a partir da página para desenvolvedores, do Facebook (Figura 41).



Geração de token

O token da página é obrigatório para começar a usar os API. O token desta página terá todas as permissões do Messenger mesmo se a tua aplicação ainda não tiver aprovação para as usar, nesse caso só vais poder enviar mensagens aos administradores da aplicação. Também podes gerar tokens de página para as páginas que não geres através do Início de sessão do Facebook.

Página: Token de acesso à página:

[Criar uma página nova](#)

Figura 41 – Acesso ao *token* da página de Facebook

Após a conclusão destes três passos referidos, a integração da Api.ai com o Facebook Messenger estava efetuada. Ao enviar mensagens através do Messenger, há uma conexão com a ferramenta Api.ai, e é enviada de volta uma resposta para a primeira plataforma.

4.1.4. Criação de *Intents*

A terceira fase passa pela criação de *intents*, que permitem definir frases que possivelmente o utilizador dirá, tentando definir um padrão ou respostas-tipo. Aqui podem ser referenciadas entidades criadas nesta ferramenta (definidos na segunda fase – subcapítulo 4.1.2), ou entidades já predefinidas. Por exemplo, existem entidades já existentes para números (*@sys.number*, *@sys.number-integer*), URL's (*@sys.url*), uma data ou períodos de tempo (*@sys.date-time*, *@sys.date-period*), relacionadas com localização (*@sys.address*, *@sys.geo-country*), entre outros. Todas estas entidades estão referenciadas na documentação desta ferramenta (Api.ai, 2016).

Vários exemplos de *intents* vão ser apresentados nos subcapítulos 4.3, 4.4, 4.5 e 4.6, referentes às funcionalidades desenvolvidas.

4.1.5. Ligação com o Servidor

Para que haja uma conexão com um servidor, que neste caso é com o *back-end* referido no subcapítulo 4.2, foi necessário criar um *webhook*, através da opção *Fulfillment* na Api.ai (Figura 42).

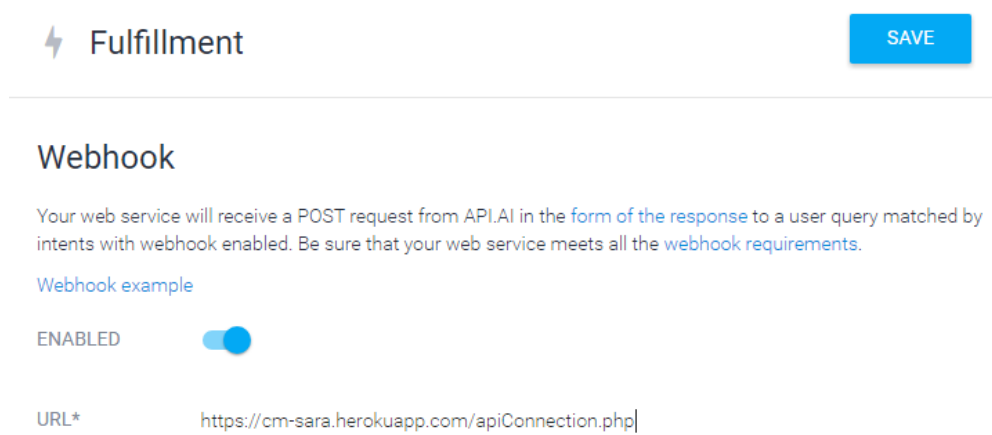


Figura 42 – Ativação do *webhook*

Nesse separador é possível ativar o *webhook*, de modo a que o *back-end* receba um pedido POST¹¹ da Api.ai na forma de um JSON, com o pedido do utilizador, que entrou num *intent*. Para além disto, apenas é se tem que incluir o URL do *back-end*. Nos *intents* nos quais é necessária uma conexão com o *back-end*, é indispensável também ativar o uso do *webhook* (Figura 43).

¹¹ Método usado para solicitação de uma resposta entre um cliente e um servidor.

The image shows a screenshot of the Google Assistant developer console. At the top, there's a 'Response' section with a question mark icon. Below it is a 'Text response' input field with a placeholder 'Enter a text response...'. Below the input field is a button labeled 'ADD MESSAGE CONTENT'. Below that is the 'Fulfillment' section. In the 'Fulfillment' section, there are two checkboxes: 'Use webhook' (which is checked) and 'Use webhook for slot-filling' (which is unchecked).

Figura 43 – Ativação do uso do *webhook* para um *intent*

Esta conexão é necessária para se conseguir dar ao utilizador uma resposta dinâmica referente ao pedido em questão.

4.2. Back-end

De modo a haver ligação entre a Api.ai e o *web service* da Casa da Música, uma vez que os pedidos do utilizador podem requerer respostas dinâmicas (consoante o artista, ou relativamente a eventos), foi necessário criar um *back-end*.

Este subcapítulo refere a criação deste mesmo *back-end*, no qual foi adotada a linguagem de programação PHP, a ligação do mesmo com o Heroku, assim como a criação de um repositório no Bitbucket, para controlo de versões deste projeto.

4.2.1. Criação de Back-end em PHP

Para que houvesse forma de o utilizador obter respostas dinâmicas, consoante os artistas e eventos, disponíveis no *web service* da Casa da Música, foi necessário criar um *back-end*. Para este, foi escolhida a linguagem de programação PHP. O seu objetivo é existir ligação entre a ferramenta Api.ai, para receber e enviar informação, assim como aceder e fazer pedidos ao *web service* da Casa da Música.

A primeira fase consiste na ligação com a Api.ai, de modo a receber um JSON, com a informação enviada pelo utilizador. Depois, esse pedido é processado neste *back-end*, e são realizados pedidos do tipo GET, ao *web service* da Casa da Música, para obter os dados pretendidos. Consoante aquilo que for devolvido por este *web service*, é definida uma resposta a ser enviada ao utilizador. A última fase consiste numa nova ligação com a Api.ai, mas desta vez para enviar os dados, também em JSON, para que a resposta seja endereçada ao utilizador.

4.2.2. Ligação com o Heroku

Para que o *back-end* estivesse alojado num servidor, de forma a ser executado pela Api.ai, foi escolhida a plataforma Heroku, para tal. Esta plataforma permite que os desenvolvedores consigam criar, executar e modificar aplicações, através da *cloud*.

Assim, tendo já criado o projeto PHP, foi criada uma aplicação no Heroku (cm-sara). De seguida foi realizado o *deploy* do projeto, para esta aplicação, seguindo os passos referentes no sítio *web* do Heroku.

Por cada vez que são feitas alterações no projeto, é realizado um *commit* com um comentário referente a essas alterações, seguindo de um *push*, que envia as alterações para o Heroku. Isto é realizado através da linha de comandos, executando os seguintes comandos:

```
git add .
git commit -am "Alterações realizadas"
git push heroku master
```

Figura 44 – Comandos para envio de alterações para o Heroku

4.2.3. Repositório Bitbucket

Uma vez que o Heroku não permite fazer um controlo de versões, foi então criado um repositório no Bitbucket, e associado este projeto.

No final de cada dia de implementação, era feito um *push*, para que as alterações criadas (*commits* realizados através dos comandos apresentados na Figura 44) fossem enviadas para o repositório. Para este processo de *push*, foi utilizada a ferramenta Tortoise Git, referida no subcapítulo 1.4. Todos os *commits* ficam disponíveis no sítio *web* do Bitbucket (Figura 45).



Figura 45 – *Commits* no repositório do Bitbucket

Este sistema é bastante útil para verificar os progressos realizados, assim como voltar a um estado anterior do projeto, caso seja necessário.

4.2.4. Receção do Pedido e Envio de Respostas

O utilizador ao comunicar com o agente, realiza constantemente vários pedidos. Esses pedidos são enviados para a ferramenta Api.ai e consequentemente associados a *intents*. Caso estes tenham ligação com o *back-end*, um JSON com toda a informação necessária é enviado para o servidor, de modo a ser obtida uma resposta.

O primeiro passo no *back-end* é verificar qual o *intent* que foi invocado (Figura 46), e depois, consoante esse pedido, é definida uma resposta. Essa resposta é posteriormente recebida de novo pela Api.ai e só depois enviada ao utilizador.

```
$intentName = $obj['result']['metadata']['intentName'];

if(strcmp($intentName, "getArtist")==0){
    ...
}else if(strcmp($intentName, "getTranslationArtist")==0){
    ...
}else if(strcmp($intentName, "getInfoEventWithoutContext")==0 || strcmp($intentName,
"getInfoEventWithContext")==0){
    ...
}else if(strcmp($intentName, "buyTicketsWithContext")==0 || strcmp($intentName,
"buyTicketsWithoutContext")==0){
    ...
}else if(strcmp($intentName, "getDateEventWithoutContext")==0 || strcmp($intentName,
"getDateEventWithContext")==0){
    ...
}else if(strcmp($intentName, "getRoomEventWithoutContext")==0 || strcmp($intentName,
"getRoomEventWithContext")==0){
    ...
}else if(strcmp($intentName, "getPriceEventWithoutContext")==0 || strcmp($intentName,
"getPriceEventWithContext")==0){
    ...
}
```

Figura 46 – Verificação do *intent* invocado

Para o envio da mensagem, foram criados alguns métodos consoante o que se queria enviar, e seguidamente serão apresentados dois deles: envio de uma mensagem simples e envio de um cartão, com hipóteses de resposta (Figura 47).

```

function sendSimpleMessage($speech, $intentName){
    $output=array();
    $output["contextOut"] = array( array( "name" => "cm",
        "lifespan"=>3, "parameters" => array( "context" => "",
        "date" => date(DATE_ATOM, mktime(0, 0, 0, 0, 0, 2017))));
    $output["speech"] = $speech;
    $output["displayText"] = $speech;
    ["source"] = $intentName;

    echo json_encode($output);
}

function sendCardRichMessage($speech, $intentName, $titleRichMessage){
    $output=array();
    $output["contextOut"] = array( array( "name" => "cm",
        "lifespan"=>3, "parameters" => array( "context" => "",
        "date" => date(DATE_ATOM, mktime(0, 0, 0, 0, 0, 2017))));
    $output["messages"] = array(array("type" => 0, "speech" => $speech),
        array("type"=> 1, "title" => $titleRichMessage,
        "buttons" => array(array("text" => "Yes", "postback" => ""),
        array("text" => "No", "postback" => ""))));
    $output["source"] = $intentName;

    echo json_encode($output);
}

```

Figura 47 – Definição das respostas (*back-end*)

É necessário definir um conjunto de parâmetros na forma de JSON, para de novo enviar para a ferramenta Api.ai. Só com estes parâmetros e com o formato apresentado na Figura 47 é que as respostas são válidas.

4.3. Compra de Bilhetes

A funcionalidade de compra de bilhetes é a mais importante deste protótipo. Para esta foi necessária ligação com o *back-end*, de forma a recolher todas as informações necessárias para dar ao utilizador. Tanto entidades foram usadas (*@nameEvent*), assim como *intents* (*buyTicketsATM*, *buyTicketsCard*, *buyTicketsCardInsertData*, *buyTicketsNo*, *buyTicketsWithContext*, *buyTicketsWithoutContext* e *buyTicketsYes*). Todos estes campos são abordados neste subcapítulo e também a forma como foi implementada esta funcionalidade.

4.3.1. Entidades

Para desenvolver esta funcionalidade, foram utilizadas várias entidades. A única entidade própria utilizada, ou seja, criada para este propósito, foi a *@nameEvent*, de forma a referenciar os nomes dos eventos da agenda cultural da Casa da Música. Depois recorreu-se ao uso de outras entidades já existentes nesta ferramenta: *@sys.given-name*, que diz respeito ao nome do utilizador, referente no cartão de crédito ou débito, *@sys.number-sequence*, que refere o número

do mesmo cartão, assim como *@sys.number-integer*, para indicar o número de segurança do cartão, *@sys.date* que referencia a data de expiração do cartão, e por último *@sys.email*, para o utilizador indicar qual o seu endereço de *email*, para que os bilhetes sejam enviados para lá.

4.3.2. Processo de Compra de Bilhetes

Caso um utilizador já tenha referenciado anteriormente o nome de um evento, esse nome é guardado num contexto. Quando o utilizador refere que pretende comprar bilhetes, o agente assume imediatamente que quer comprar bilhetes para o evento que referiu. Para tal, foi criado o *intent buyTicketsWithContext* (Figura 48).

buyTicketsWithContext SAVE

Contexts

getinfoeventc ⓧ Add input context

4 getinfoeventc ⓧ Add output context ✕

User says Search in user says 🔍

” Add user expression

” buy 8 tickets

” tickets

” I want tickets

Events ?

Action

Enter action name

| REQUIRED ? | PARAMETER NAME ? | ENTITY ? | VALUE | IS LIST ? | PROMPTS ? |
|-------------------------------------|-------------------------------|-----------------------|--------------------------|--------------------------|------------------------|
| <input checked="" type="checkbox"/> | nameEvent | @nameEvent | #getinfoeventc.nameEvent | <input type="checkbox"/> | Can you tell me... |
| <input checked="" type="checkbox"/> | numberTickets | @sys.number-integer | \$numberTickets | <input type="checkbox"/> | And how many ti... |

Figura 48 - *Intent buyTicketsWithContext*

Como é possível verificar na Figura 49, foram pedidas informações relativas a um evento, e quando o utilizador disse que pretendia adquirir bilhetes, o agente já sabia qual o evento que este se estava a referir.

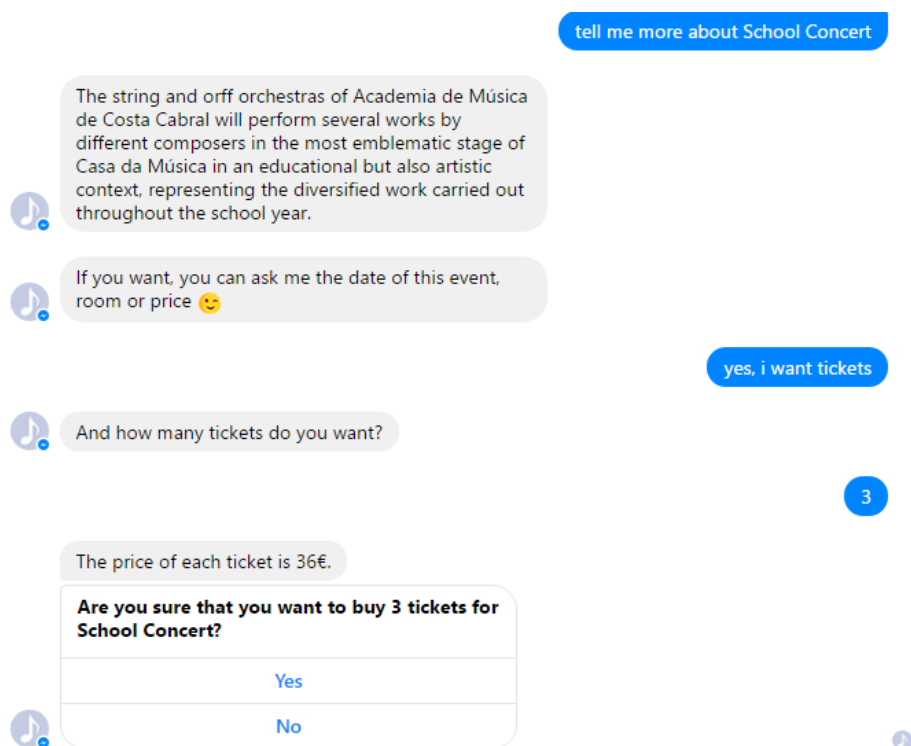
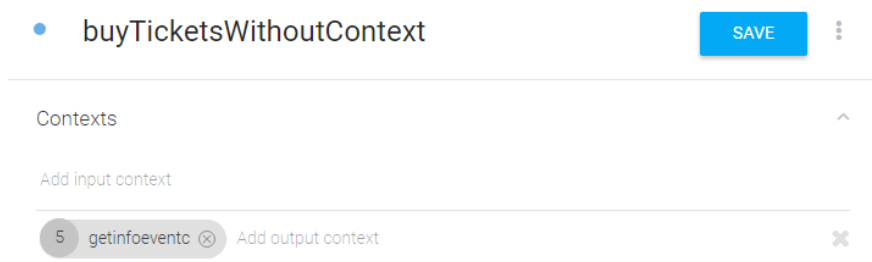


Figura 49 – Pedido de bilhetes com contexto

Por outro lado, se o utilizador ainda não referiu nenhum evento e diz que quer adquirir bilhetes, é necessário que este indique qual o nome do evento. Se indicar “Quero comprar bilhetes para o evento X”, já está a referir o evento, mas caso diga “Quero comprar bilhetes”, o agente pergunta qual o nome. Aqui é realizado um pedido de compra de bilhetes, sem contexto (*intent buyTicketsWithoutContext* - Figura 50).

Figura 50 - *Intent buyTicketsWithoutContext*

Já neste caso, a Figura 51, permite verificar qual o fluxo de interações entre o agente e o utilizador.

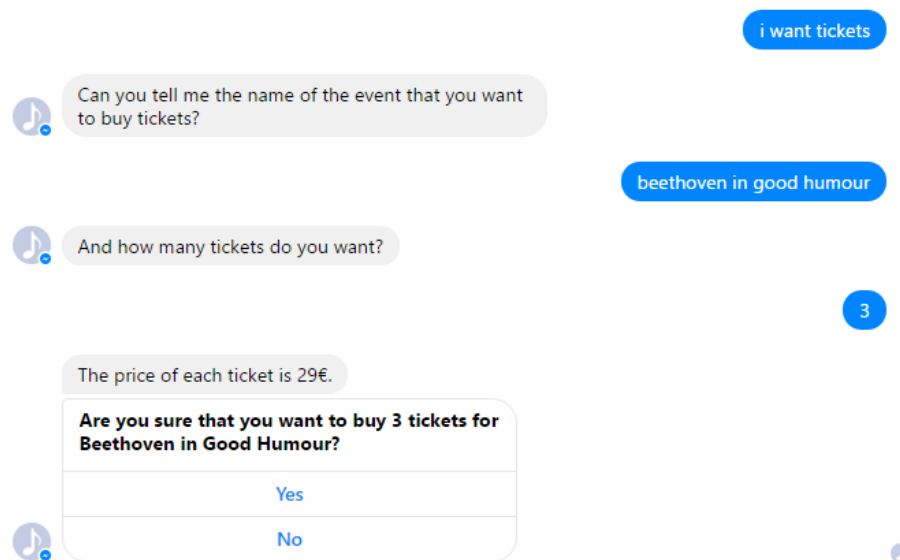


Figura 51 – Compra de bilhetes sem contexto

Tanto o *intent buyTicketsWithContext*, como o *buyTicketsWithoutContext* têm um contexto de saída (*getinfoeventc*) para enviarem a informação necessária para o *intent* seguinte, consoante o fluxo estipulado. Já o primeiro tem também um contexto de entrada para receber o nome do evento que já foi mencionado anteriormente.

Se o utilizador tiver referido um evento numa conversa anterior e disser que quer comprar bilhetes para outro, é ativado um segundo contexto, relativo ao segundo evento (Figura 52).

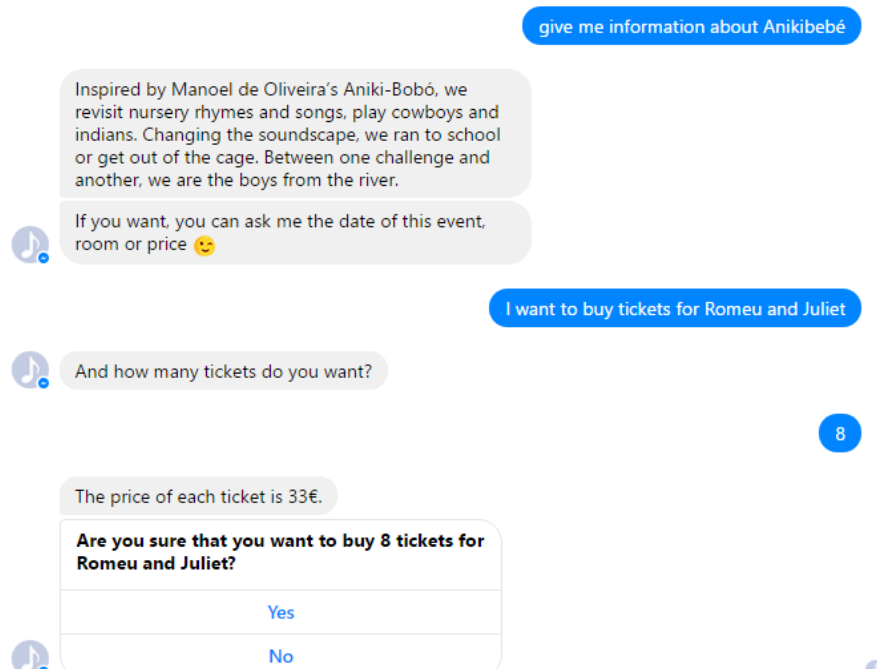


Figura 52 - Compra de bilhetes para um evento específico

É perguntado ao utilizador qual o número de bilhetes que pretende e calculado o preço através do *back-end*. Após esse cálculo, é enviada essa informação e passa a ser necessária uma

confirmação da parte do utilizador, quanto ao prosseguimento da compra de bilhetes (*intents buyTicketsNo* e *buyTicketsYes*). Caso este confirme, é requerida a escolha da forma de pagamento que pretende (Figura 53): cartão de crédito/débito ou terminal multibanco (*intents buyTicketsATM*, *buyTicketsCard* e *buyTicketsCardInsertData*).

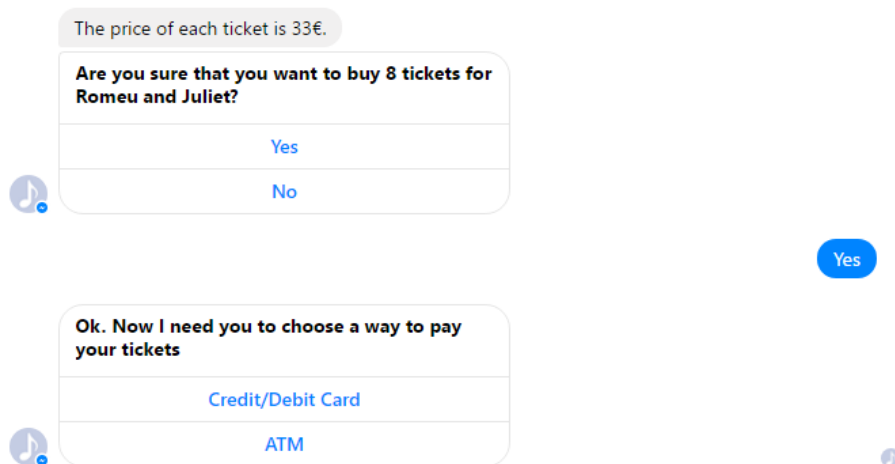


Figura 53 – Mensagem de escolha da forma de pagamento

4.3.3. Formas de Pagamento

O utilizador pode escolher duas formas de pagamento: cartão de crédito/débito (*intents buyTicketsCard* e *buyTicketsCardInsertData*), ou terminal multibanco (*intent buyTicketsATM*).

No primeiro caso, o utilizador necessita inserir todos os dados referentes ao cartão: nome, número, código de segurança, data de expiração. E em ambos os casos, é necessária a introdução do endereço de *email* para que os bilhetes, após o pagamento realizado, sejam enviados.

O *intent buyTicketsCard* apenas serve para o agente referir que o utilizador deve indicar os dados do seu cartão de crédito ou débito. Quando este os envia ao agente, através do Facebook Messenger, é ativado o *intent buyTicketsCardInsertData* (Figura 54), que possibilita o controlo de que todos os dados necessários foram realmente enviados.

• buyTicketsCardInsertData SAVE

Events ?

Action

Enter action name

| REQUIRED | PARAMETER NAME | ENTITY | VALUE | IS LIST | PROMPTS |
|-------------------------------------|----------------|----------------------|------------------|--------------------------|-------------------|
| <input checked="" type="checkbox"/> | name | @sys.given-name | \$name | <input type="checkbox"/> | I need your fi... |
| <input checked="" type="checkbox"/> | cardNumber | @sys.number-sequence | \$cardNumber | <input type="checkbox"/> | Ok :) Please i... |
| <input checked="" type="checkbox"/> | securityCode | @sys.number-integer | \$securityCode | <input type="checkbox"/> | Now I need y... |
| <input checked="" type="checkbox"/> | expirationDate | @sys.date | \$expirationDate | <input type="checkbox"/> | Can you plea... |
| <input checked="" type="checkbox"/> | email | @sys.email | \$email | <input type="checkbox"/> | And the last t... |
| <input type="checkbox"/> | Enter name | Enter entity | Enter value | <input type="checkbox"/> | — |

Figura 54 – Intent buyTicketsCardInsertData

Se o utilizador não inserir os dados necessários, o agente pede-lhe o que está em falta (Figura 55).

Ok. Now I need you to choose a way to pay your tickets

Credit/Debit Card

ATM

Credit/Debit Card

So now, I need your card details 😊

My card name is Sara Oliveira, number 1234 1234 1234 1234 and security code 123

Can you please tell me the expiration date of your card (DD-MM-YYYY) ? 😊

21-06-2017

And the last thing I need, is your email address 😊

so9941@hotmail.com

Thanks for all 😊 You will receive an email with your tickets 😊

Figura 55 – Compra de bilhetes (cartão de crédito/débito)

O intent buyTicketsATM (Figura 56) tem o propósito de quando o utilizador escolhe a opção de pagamento através do terminal multibanco, pedir-lhe o endereço de *email*, e enviar-lhe os dados para pagamento através da opção escolhida.

• buyTicketsATM SAVE

Contexts ^

buyticketsyes ⓧ Add input context

Add output context ✕

User says Search in user says 🔍 ^

” Add user expression

” ATM

Events ? ∨

Action ^

Enter action name

| REQUIRED ? | PARAMETER NAME ? | ENTITY ? | VALUE | IS LIST ? | PROMPTS ? |
|-------------------------------------|-------------------------------|-----------------------|-------------|--------------------------|------------------------|
| <input checked="" type="checkbox"/> | email | @sys.email | Enter value | <input type="checkbox"/> | Can you please ... |
| <input type="checkbox"/> | priceTicket | Enter entity | #cm.value | <input type="checkbox"/> | — |

Figura 56 - Intent *buyTicketsATM*

A compra de bilhetes através da segunda opção de pagamento, processa-se como descrito, e um exemplo é apresentado na Figura 57.

ATM

Can you please tell me your email address? 😊

so9941@hotmail.com

This is the information you need to pay 😊
 Entity: 10000
 Reference: 123 123 123
 Value: 123 €

After the payment is finished, you will receive an email 😊 Thanks for all 😊

Figura 57 - Compra de bilhetes (ATM)

Todos os *intents* referenciados neste subcapítulo e no anterior (4.3.2), recorrem também a contextos para que o nome do evento, preço e número de bilhetes que o utilizador pretende adquirir, estejam disponíveis em todos os *intents* relativos à compra de bilhetes.

4.4. Obter Informações sobre Eventos

A funcionalidade de obtenção de informações sobre eventos, como requer resultados dinâmicos, necessita de ligação com o *back-end*.

Foi necessária a criação de entidades (*@nameEvent*) e *intents* (*getDateEvent*, *getDateEventWithContext*, *getInfoEvent*, *getInfoEventWithContext*, *getPriceEventWithContext*, *getPriceEventWithoutContext*, *getRoomEvent*, *getRoomEventWithContext*). Ambos estes pontos serão abordados neste subcapítulo, assim como os tipos de pedido que o utilizador pode realizar, relativos a detalhes informacionais sobre o evento em questão, mas também relativos à data/hora, sala onde ocorre o evento e preço dos bilhetes.

Também as respostas dadas ao utilizador, consoante cada tipo de pedido, e tendo contextos associados ou não, serão também apresentadas neste subcapítulo.

4.4.1. Entidades

Para esta funcionalidade foi criada apenas uma entidade (*@nameEvent*), que é a mesma que foi utilizada na funcionalidade de compra de bilhetes. Aqui foram inseridos os eventos que o utilizador pode obter informações. Só assim a ferramenta Api.ai, através dos *intents* criados (*getDateEvent*, *getDateEventWithContext*, *getInfoEvent*, *getInfoEventWithContext*, *getPriceEventWithContext*, *getPriceEventWithoutContext*, *getRoomEvent*, *getRoomEventWithContext*), é que consegue reconhecer os eventos aos quais os utilizadores se referem. É possível inserir vários nomes de eventos através de um ficheiro, o que seria útil num caso real. Para não se inserir manualmente, poderia ser feita uma exportação dos nomes dos eventos através do *web service* da Casa da Música, e depois estes serem inseridos na Api.ai através do ficheiro referido. Contudo, como se trata de um protótipo, apenas foram inseridos manualmente alguns nomes de eventos possíveis.

4.4.2. Tipos de Pedido

Visto que dentro da funcionalidade de obtenção de informação sobre eventos, é possível obter detalhes, informação relativa à hora/data, sala ou preço de um determinado evento, neste subcapítulo serão abordadas como foram implementadas estas quatro hipóteses de pedidos.

4.4.2.1. Detalhes

Um dos tipos de pedido que o utilizador pode realizar, é pedir detalhes informacionais sobre um determinado evento. Para tal, foi necessária a criação de dois *intents*

(*getInfoEventWithoutContext* e *getInfoEventWithContext*), apresentado nas duas figuras abaixo (Figura 58 e Figura 59).

The screenshot shows the Google Assistant configuration page for the intent **getInfoEventWithoutContext**. At the top right is a blue **SAVE** button. Below the title, there is a section for **Contexts** with an upward arrow. Under this section, there is a field for **Add input context** and a list of contexts. The first context is **getinfoeventc** with a close icon (X). Below this is a field for **Add output context**. Further down is a section for **User says** with a search bar labeled **Search in user says** and a magnifying glass icon. Below the search bar is a list of user expressions: "Add user expression", "can you give me information about nameEvent?", "tell me more about nameEvent", and "give me details about name event".

Figura 58 – *Intent getInfoEventWithoutContext*

Ao primeiro *intent* está associado um contexto de saída (*getinfoeventc*), para se o utilizador quiser comprar bilhetes ou se quiser pedir a data/hora, na qual esse evento decorre, não ter de referir novamente o nome do mesmo.

The screenshot shows the Google Assistant configuration page for the intent **getInfoEventWithContext**. At the top right is a blue **DONE** button. Below the title, there is a section for **Contexts** with an upward arrow. Under this section, there is a field for **Add input context** and a list of contexts. The first context is **getinfoeventc** with a close icon (X). Below this is a field for **Add output context**. Further down is a section for **User says** with a search bar labeled **Search in user says...** and a magnifying glass icon. Below the search bar is a list of user expressions: "Add user expression", "can you give me details about that event?", "info", and "tell me information about that".

Figura 59 – *Intent getInfoEventWithContext*

Já ao segundo, para além do contexto de saída, está também associado um contexto de entrada, uma vez que o nome do evento já foi referenciado anteriormente e este se encontra nesse

mesmo contexto. Assim, o utilizador não tem de repetir novamente o nome do evento, caso o tenha referido anteriormente.

Em ambos os *intents* foram definidas também algumas frases que o utilizador poderá dizer, mas não significa que tenha de dizer apenas estas. São só algumas possibilidades que dão informação de aprendizagem ao agente.

No *intent getInfoEventWithContext*, o parâmetro *nameEvent* foi definido como obrigatório, e caso não tenha nenhum valor associado, é realizada uma pergunta ao utilizador (Figura 60).

The screenshot shows a user interface for defining prompts for the 'nameEvent' entity. At the top, there is a header 'Prompts for "nameEvent"' with a close button (X). Below this is a table with three columns: NAME, ENTITY, and VALUE. The table contains one row with the values 'nameEvent', '@nameEvent', and '\$nameEvent' respectively. Below the table, there is a section titled 'PROMPTS' with a list of two prompts: '1 Can you tell me the name of the event please? :)' and '2 Enter a prompt variant...'.

| NAME | ENTITY | VALUE |
|-----------|------------|-------------|
| nameEvent | @nameEvent | \$nameEvent |

PROMPTS

- 1 Can you tell me the name of the event please? :)
- 2 Enter a prompt variant...

Figura 60 – Pergunta definida para o utilizador dar o nome do evento

4.4.2.2. Data/Hora

O outro tipo de pedido possível, é referente à data/hora de um determinado evento. Foram necessários dois *intents* (*getDateEventWithoutContext* e *getDateEventWithContext*), em que no primeiro (Figura 61) é utilizada a entidade criada (*@nameEvent*), nas frases do utilizador, enquanto que no segundo (Figura 62) não, porque o nome do evento já é conhecido e vem no contexto de entrada.

• getDateEventWithoutContext SAVE ⋮

Contexts ^

Add input context

3 getinfoevento (X) Add output context X

User says Search in user says 🔍 ^

” Add user expression

” Give me the date

” Can you give me the date of nameEvent?

” When is nameEvent

” I want to know when will be nameEvent

Figura 61 – Intent getDateEventWithoutContext

• getDateEventWithContext SAVE ⋮

Contexts ^

getinfoevento (X) Add input context

3 getinfoevento (X) Add output context X

User says Search in user says 🔍 ^

” Add user expression

” give me the date

” when?

” when is that event?

” hours

” date

” tell me the date

Figura 62 – Intent getDateEventWithContext

Em ambos os *intents* foram definidas mais uma vez, algumas possibilidades de frases que o utilizador dirá ao requerer este tipo de informação. No primeiro caso apenas há um contexto de saída, mas no segundo há dois contextos, um de entrada e outro de saída. Caso o nome do evento já tenha sido referido, o pedido entra no *intent getDateEventWithContext*, caso contrário, entra no *getDateEventWithoutContext*.

Tal como foi referido no subcapítulo anterior (4.4.2.1), também no *intent getDateEventWithoutContext*, o parâmetro *nameEvent* foi definido como obrigatório.

4.4.2.3. Sala

O utilizador também pode pedir ao agente qual a sala onde um evento se vai realizar. Mais uma vez, pode haver um contexto (*intent getRoomEventWithContext*), ou não (*intent getRoomEventWithoutContext*). No caso de não haver contexto, é adicionado um contexto de entrada (Figura 63).

• **getRoomEventWithContext** SAVE

Contexts

getinfoevento × Add input context

3 getinfoevento × Add output context

User says Search in user says Q

“ Add user expression

“ give me the room

“ hum... can you please tell me the room?

“ can you please tell me the room?

“ room

“ Can you tell me the room please?

Events ?

Action

Enter action name

| REQUIRED ? | PARAMETER NAME ? | ENTITY ? | VALUE | IS LIST ? |
|--------------------------|-------------------------------|-----------------------|--------------------------|--------------------------|
| <input type="checkbox"/> | nameEvent | @nameEvent | #getinfoevento.nameEvent | <input type="checkbox"/> |
| <input type="checkbox"/> | Enter name | Enter entity | Enter value | <input type="checkbox"/> |

Figura 63 – *Intent getRoomEventWithContext*

Ambos os *intents* necessitam também de uma ligação com o *back-end*, de forma ao agente conseguir obter a sala do evento em questão.

4.4.2.4. Preço

Por último, é possível também que o agente devolva ao utilizador o preço de um evento. Este pode já estar contextualizado (*intent getPriceEventWithContext*) ou não (*intent*

getPriceEventWithoutContext - Figura 64). Os contextos de entrada e de saída processam-se da mesma forma que em todas as funcionalidades anteriores.

getPriceEventWithoutContext SAVE

Contexts

Add input context

3 getinfoevento Add output context

User says

Search in user says

add user expression

price

I want the price of nameEvent

Events

Action

Enter action name

| REQUIRED | PARAMETER NAME | ENTITY | VALUE | IS LIST | PROMPTS |
|-------------------------------------|----------------|--------------|-------------|--------------------------|--------------------|
| <input checked="" type="checkbox"/> | nameEvent | @nameEvent | \$nameEvent | <input type="checkbox"/> | Which is the na... |
| <input type="checkbox"/> | Enter name | Enter entity | Enter value | <input type="checkbox"/> | — |

Figura 64 – Intent *getPriceEventWithoutContext*

Mais uma vez, para se determinar qual o preço de um evento, é necessária uma conexão com o *back-end*. Contudo, esta informação não estava disponível através do *web service* da Casa da Música. Todos os eventos referem um preço de -1. Assim sendo, para simular este valor, foi definida uma fórmula para dar ao utilizador este dado (50-dia do evento), e devolvida através do método *getPrice* (Figura 67).

```
function getPrice($obj){
    $dateTime = getDateTime($obj);
    $day = date('d', $dateTime[2]);
    $price = (50-$day);

    return $price;
}
```

Figura 65 – Método de devolução do preço de um evento

A função *getDateTime(\$obj)* é também uma função criada, para retorno da data e horas de um evento e também comunica com o *web service* da Casa da Música.

4.4.3. Respostas

Dentro desta funcionalidade, é possível obter detalhes sobre um evento, com ou sem contexto, assim como informações relativas à hora/data do mesmo, sala e preço dos bilhetes, novamente

com ou sem contexto. Alguns exemplos de respostas a pedidos que podem ser feitos pelos utilizadores, são seguidamente fornecidos.

O primeiro caso, é se um evento não for encontrado através do *back-end*, que recorre ao *web service* da Casa da Música (Figura 66).

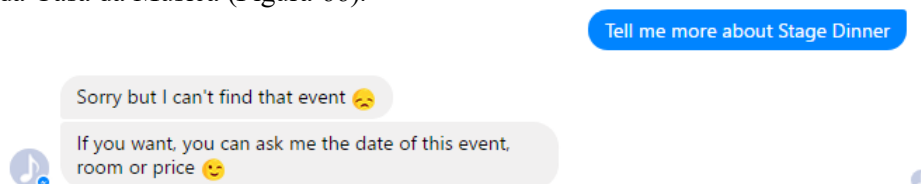


Figura 66 – Informações sobre um evento não encontrado

No caso apresentado na Figura 67, é feito um pedido de informação sobre um evento, inicialmente relativo a detalhes, sem contexto, e depois relativo a horas/data, já com contexto.

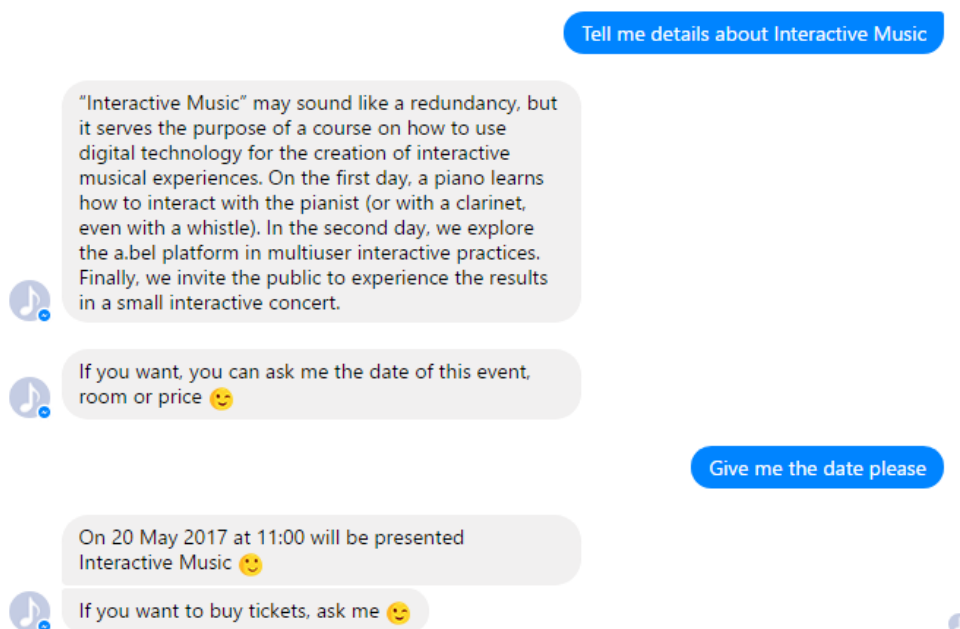


Figura 67 – Pedido de informações e data de um evento (I)

Já na Figura 68, é feito o inverso. O utilizador pergunta quando é que o evento se realiza, ou seja, é feito um pedido relativo a data/horas, sem contexto, e de seguida um pedido relativo a detalhes, já com contexto.

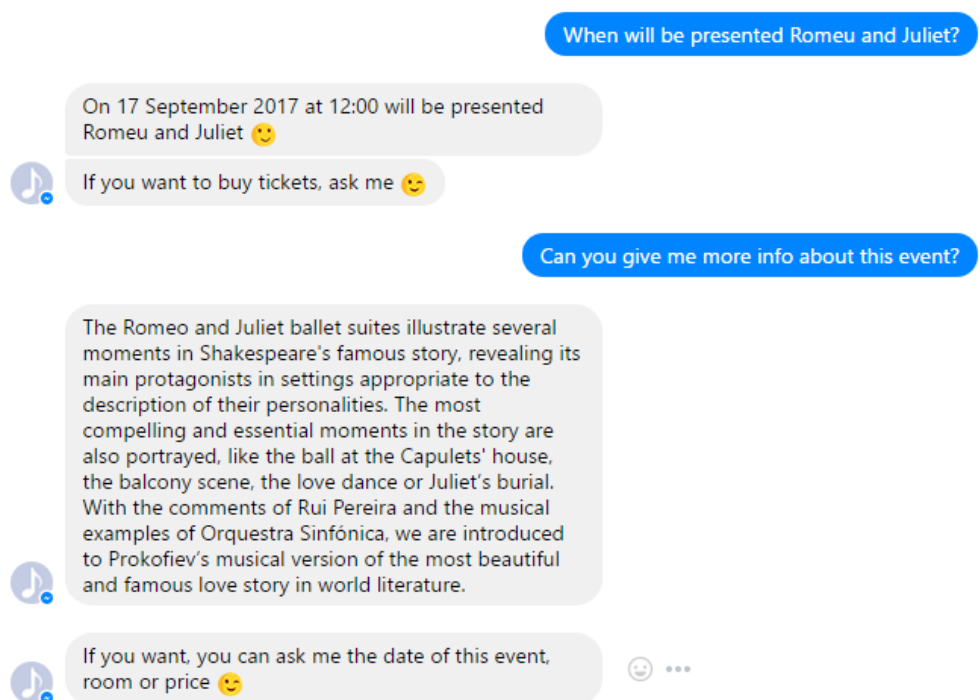


Figura 68 - Pedido de informações e data de um evento (II)

Foi definida uma pergunta nos *intents* *getInfoEventWithContext* e *getDateEventWithoutContext* (subcapítulos 4.4.2.1 e 4.4.2.2), caso o pedido feito pelo utilizador não esteja completo. Esse caso é apresentado na Figura 69.

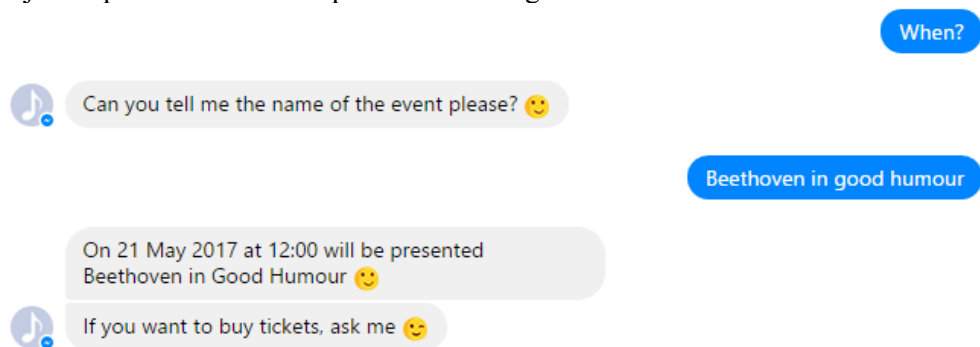


Figura 69 - Pedidos de data de um evento (sem contexto)

A Figura 70 apresenta um exemplo, caso o utilizador pretenda saber a sala de um evento (sem contexto), onde é questionado o nome do mesmo. Depois é realizado um pedido relativo ao preço de cada bilhete, e não é questionado o nome do evento, porque este está referenciado no contexto do *intent* ativado (*getPriceEventWithContext*).

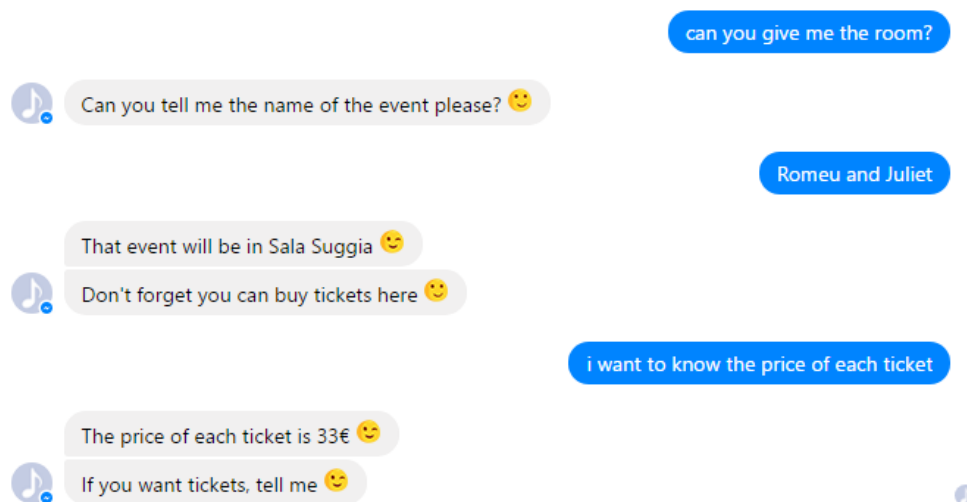


Figura 70 - Pedidos de informação sobre a sala e preço de um evento

Em todas estas opções, são dadas ao utilizador algumas sugestões relativas ao que este pode fazer, conversando com o agente.

4.5. Obter Informações sobre Artistas

Esta funcionalidade necessitou de ligação com o *back-end*, uma vez que as respostas variam consoante o artista em questão, gerando assim respostas dinâmicas.

Todas as entidades utilizadas (*@sys.music-artist*, *@artist*, *@sys.given-name* e *@sys.last-name*), do sistema ou não, serão abordadas neste subcapítulo, tendo em conta alguns tipos de experimentação realizados, que falharam.

Também os tipos de resposta que podem ser dados ao utilizador serão aqui abordados.

4.5.1. Entidades

Para esta funcionalidade, foi necessária a criação de uma entidade (*@artist*). Inicialmente foi utilizada a entidade do sistema, *@sys.music-artist*, mas rapidamente foi abandonada esta hipótese. Ao pedir informações relativas a um artista em específico, esta entidade não reconhecia vários artistas, o que levava à permanente resposta ao utilizador, de que não tinham sido encontradas informações sobre os mesmos.

De seguida, foi criada uma entidade própria (*@artist*), onde foram inseridos alguns artistas. A Figura 38 permite visualizar essa entidade. Assim, para que a *Api.ai* reconheça um artista, é necessário que este esteja inserido na lista da entidade *@artist*. Podia ter sido feita uma importação de todos os artistas devolvidos através do *web service* da Casa da Música nesta entidade, através de um ficheiro. Caso este não fosse um protótipo, esta importação deveria ser realizada rotineiramente para haver constante atualização.

4.5.2. Tipos de Resposta

Neste caso de uso, estão envolvidos vários tipos de resposta que podem ser dadas ao utilizador que faz o pedido.

O primeiro tipo é onde são devolvidos mais detalhes relativos ao artista em questão (Figura 71).

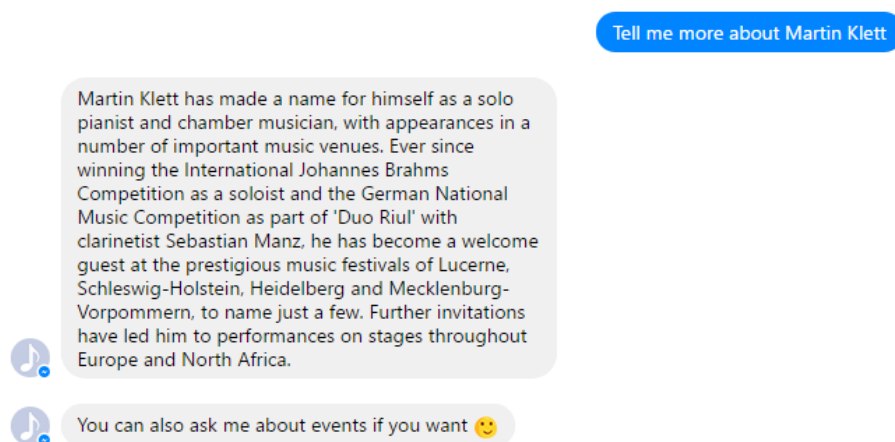


Figura 71 – Informação sobre artistas (tipo de resposta I)

Caso haja uma descrição em inglês de um artista, esta por defeito é bastante longa, levando a que a mensagem enviada ao utilizador, fosse dividida em várias respostas (Figura 72), pois excedia o limite de caracteres suportado pelo Facebook Messenger.

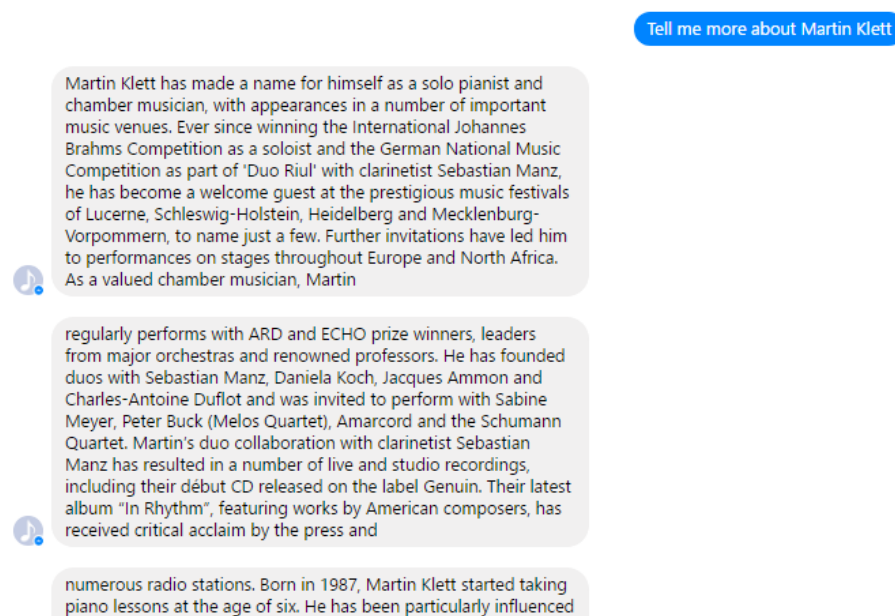


Figura 72 – Mensagem cortada em várias respostas

Para além disto acontecer, seria também uma resposta bastante longa para devolver ao utilizador, na plataforma Facebook Messenger. Assim sendo, foi criado um método (Figura 73) que delimita a mensagem até aos 600 caracteres, que é aproximadamente aquilo que é aceite para

a mensagem não se dividir. Após esta fase a mensagem (até aos 600 caracteres) é cortada até ao último ponto final que encontra.

```
function cutMessage($speech){
    $cutSpeech = substr($speech, 0, 600);
    $cutSpeech = substr($cutSpeech, 0, strrpos ( $cutSpeech, '.' ) +1);
    return $cutSpeech;
}
```

Figura 73 - Método que delimita o número de caracteres de uma mensagem

Assim, a mensagem enviada ao utilizador, referente à informação de artistas, passa a ser como a apresentada na Figura 71.

É possível também que o artista não seja encontrado, e por isso é devolvida uma resposta dizendo isso mesmo (Figura 74).

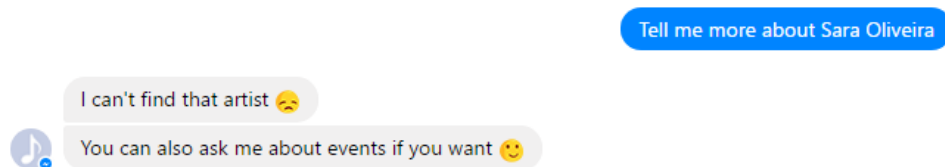


Figura 74 - Informação sobre artistas (tipo de resposta II)

Caso o artista seja encontrado, através do *web service* da Casa da Música, mas não tenha qualquer informação extra sobre o mesmo, é devolvida uma resposta indicando que não há detalhes (Figura 75).

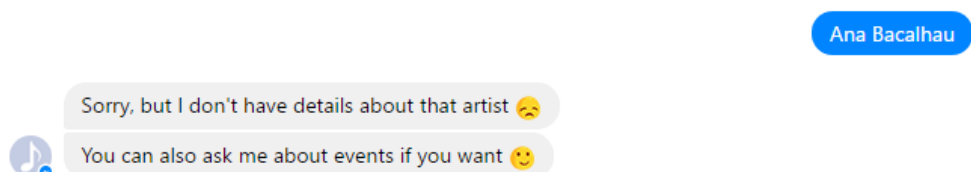


Figura 75 - Informação sobre artistas (tipo de resposta III)

Por último, é também possível que os artistas não tenham informação no idioma deste protótipo. Assim sendo, é enviada uma resposta indicando que não existem detalhes em inglês (Figura 76). Contudo, caso haja a informação em português, é enviada ao utilizador a possibilidade deste obter uma tradução automática através do Google Translate (abrindo um novo separador).

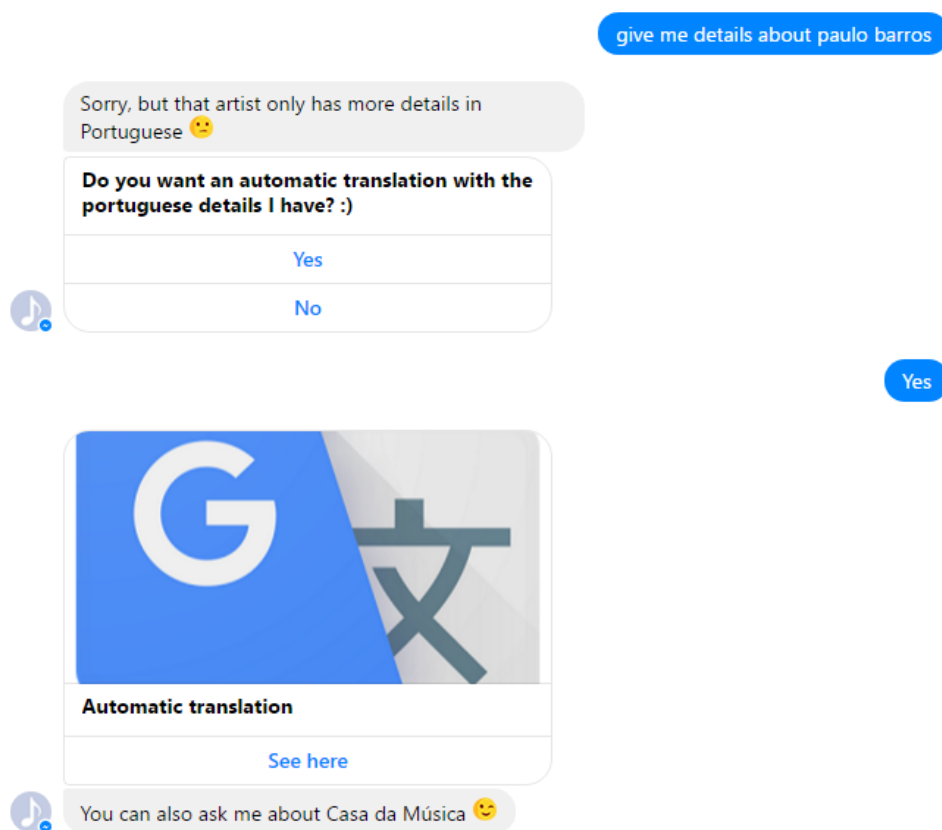


Figura 76 - Informação sobre artistas (tipo de resposta IV)

Nesta funcionalidade, independentemente do tipo de resposta, também são sempre sugeridas outras funcionalidades que o utilizador pode ter acesso através da conversa com o agente.

4.6. Obter Informações sobre a Casa da Música

Esta funcionalidade é a única que não necessita de uma ligação com o *back-end*, uma vez que as respostas a fornecer ao utilizador não são dinâmicas. Isto significa que independentemente do que o utilizador pede ao agente, não há muito a variar. No caso dos artistas por exemplo, o utilizador pode pedir informações relativas a vários artistas, mas neste caso, apenas vai pedir informações sobre uma única instituição. As respostas serão sempre as mesmas, enquanto que os artistas têm várias descrições possíveis.

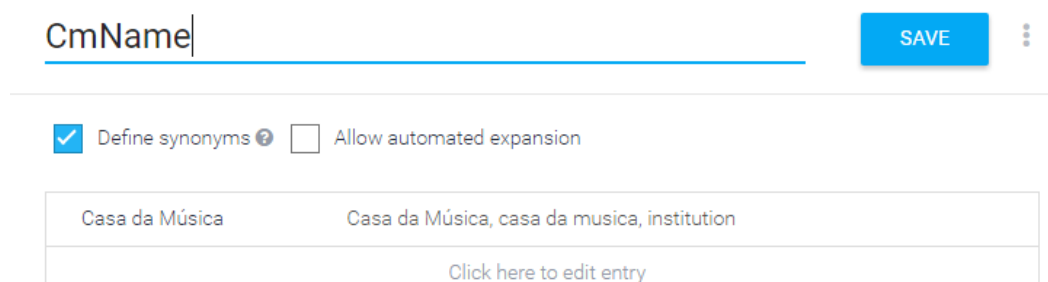
O utilizador pode pedir detalhes históricos sobre a instituição, a morada, incluindo instruções de como ir até lá (de metro, autocarro ou através do próprio carro), ou pedir o horário.

Assim sendo, foi necessária a criação de entidades (*@CmName*) e *intents* (*getAddressCM*, *getHelpCM_bus*, *getHelpCM_car*, *getHelpCM_metro*, *getHelpCM_no*, *getHelpCM_yes*, *getInfoCM* e *getScheduleCM*), de modo a ser possível este diálogo.

Inicialmente será apresentada a entidade criada, e para cada subfuncionalidade relativa ao pedido de informação sobre a instituição, será dedicado um subcapítulo para apresentação de mais detalhe.

4.6.1. Entidades

Para esta funcionalidade apenas foi necessária uma entidade (*@CmName*), que refere as possíveis palavras que podem despoletar uma ação nos *intents* referentes ao pedido de informações relativas à Casa da Música (Figura 77).



The screenshot shows the Google Assistant interface for defining an entity. At the top, the entity name 'CmName' is entered in a text field, followed by a blue 'SAVE' button and a vertical ellipsis menu. Below this, there are two checkboxes: 'Define synonyms' (checked) and 'Allow automated expansion' (unchecked). Underneath these options is a table with two columns. The first column contains the text 'Casa da Música'. The second column contains the text 'Casa da Música, casa da musica, institution'. Below the table, there is a link that says 'Click here to edit entry'.

| | |
|----------------|---|
| Casa da Música | Casa da Música, casa da musica, institution |
|----------------|---|

[Click here to edit entry](#)

Figura 77 – Entidade *CmName*

Esta entidade foi usada na definição de frases que o utilizador poderá dizer no pedido desta funcionalidade. Essas frases foram definidas nos *intents* apresentados nos subcapítulos seguintes.

4.6.2. Tipos de Pedido

Ao obter informações sobre a Casa da Música, o utilizador pode escolher se quer obter detalhes históricos sobre a instituição, morada e como ir para o local, ou então sobre o horário. Todos estes tipos de pedido por parte do utilizador, são abordados neste subcapítulo.

4.6.2.1. Detalhes Históricos

Para este tipo de pedido relativo à obtenção de informação sobre a Casa da Música, foi criado o primeiro *intent* que diz respeito aos detalhes históricos da Casa da Música (*getInfoCM*) - Figura 78.

• getInfoCM SAVE

Contexts ▼

User says Search in user says... 🔍 ^

🗣️ Add user expression

🗣️ Info about Casa da Música

🗣️ Information about Casa da Música

🗣️ tell me some historical details about Casa da Música

🗣️ Give me more details about Casa da Música

🗣️ Tell me more about Casa da Música

Events ? ▼

Action ^

Enter action name...

| REQUIRED ? | PARAMETER NAME ? | ENTITY ? | VALUE | IS LIST ? |
|--------------------------|------------------|-----------------|----------------|--------------------------|
| <input type="checkbox"/> | CmName | @CmName | \$CmName | <input type="checkbox"/> |
| <input type="checkbox"/> | Enter name... | Enter entity... | Enter value... | <input type="checkbox"/> |

Figura 78 – Intent *getInfoCM*

Aqui foram definidas algumas frases que o utilizador pode dizer para ativar uma resposta a este pedido. À partida, sempre que o utilizador pedir detalhes sobre a instituição, vai utilizar na frase as palavras indicadas na entidade *@CmName*. Há três respostas possíveis para este pedido, mas este pode ser realizado de várias formas. Nas figuras abaixo (Figura 79, Figura 80 e Figura 81) é possível verificar o referido.

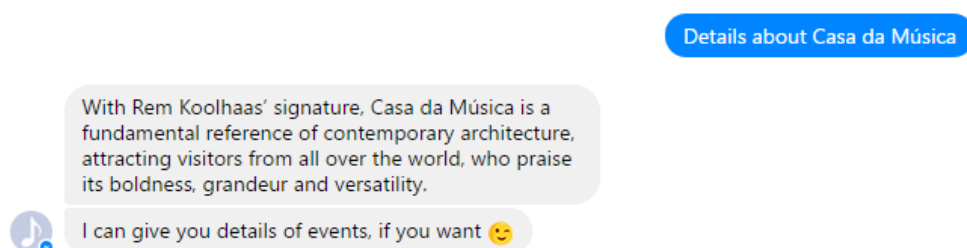


Figura 79 – Pedido de detalhes sobre a Casa da Música (I)

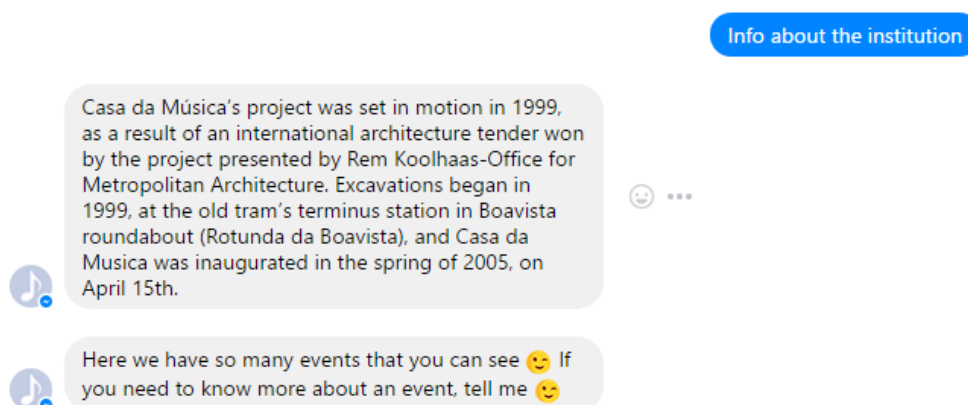


Figura 80 - Pedido de detalhes sobre a Casa da Música (II)

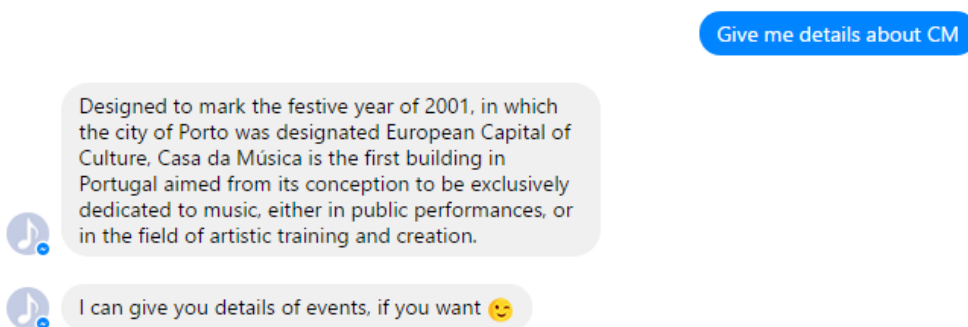


Figura 81 - Pedido de detalhes sobre a Casa da Música (III)

Como é possível verificar, nem todos os pedidos realizados pelo utilizador, estão definidos no *intent getInfoCM*, e mesmo assim a Api.ai é suficientemente inteligente para apresentar uma resposta. Todas estas respostas são enviadas aleatoriamente, assim como também as sugestões de outras possíveis funcionalidades.

4.6.2.2. Morada

O utilizador pode também requerer a morada da instituição, e para isso pode apenas perguntar onde se localiza, ou até mesmo qual é a morada da Casa da Música, por exemplo. Para tal, foi então criado o *intent getAddressCM*, onde mais uma vez foram definidas algumas possíveis entradas do utilizador.

Foram definidas duas possíveis respostas e de seguida uma pergunta do tipo *card*, que questiona ao utilizador se quer ou não indicação de como se dirigir à instituição (Figura 82).

Response ?

Text response

1 I will give you the address: Av. da Boavista, 604-610. 4149-071, Porto Portugal ;)

2 Yes, I will help you :) This is the address: Av. da Boavista, 604-610. 4149-071, Porto Portugal.

3 Enter a text response variant...

Card

Enter image URL...

Do you need to know how to get here? ;)

Enter card subtitle...

Yes

No

Enter new button title...

ADD MESSAGE CONTENT

Figura 82 – Respostas a dar ao utilizador após pedido de morada da instituição

As respostas possíveis são *Yes* ou *No*, e apresentadas da seguinte forma no Facebook Messenger (Figura 83).

give me the address

I will give you the address: Av. da Boavista, 604-610. 4149-071, Porto Portugal 😊

Do you need to know how to get here? ;)

Yes

No

Figura 83 – Pedido da morada

Como num mesmo *intent* não se podem definir fluxos decisivos entre duas ou mais respostas, foi necessário criar dois *intents* para cada resposta – *getHelpCM_no* e *getHelpCM_yes*. Para que houvesse um seguimento de um *intent* para cada um destes, foi definido um contexto de saída em *getAddressCM* (Figura 84), para ser recebido como entrada nas duas variantes, ou seja, nestes dois *intents* (Figura 85 e Figura 86). Apenas se o utilizador disser *Yes* ou *No*, e se receber este contexto, é que entra num dos dois *intents*.

• getAddressCM SAVE

Contexts ^

Add input context

1 getHelpCM Add output context ✕

Figura 84 – Contextos de *getAddressCM*

• getHelpCM_yes SAVE

Contexts ^

getHelpCM Add input context

1 getHelpCM_yes Add output context ✕

Figura 85 – Contextos de *getHelpCM_yes*

• getHelpCM_no SAVE

Contexts ^

getHelpCM Add input context

Add output context ✕

Figura 86 – Contextos de *getHelpCM_no*

No caso de o utilizador aceitar esta ajuda do agente (*Yes*), e por isso é que o *intent* *getHelpCM_yes* tem um contexto de saída (Figura 85), é apresentado outro cartão com três possibilidades de indicação do caminho até à Casa da Música: através de metro, autocarro ou o próprio carro (Figura 87).

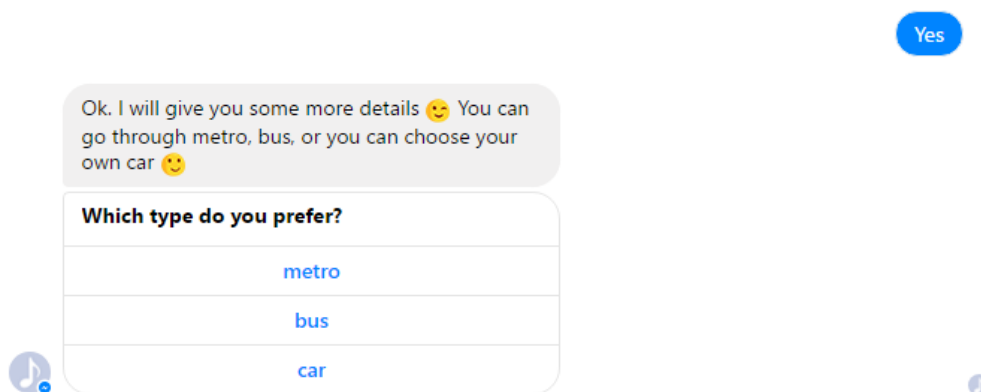


Figura 87 – Opção de veículo para indicar ao utilizador como chegar à instituição

Para cada escolha do utilizador, consoante as três possibilidades disponibilizadas, o agente devolve a informação pretendida (Figura 88, Figura 89 e Figura 90).

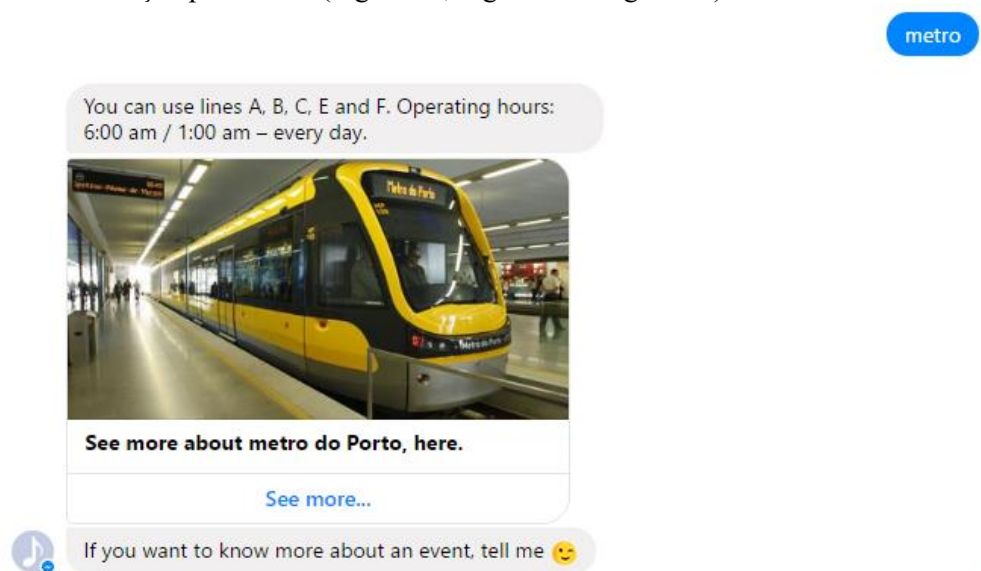


Figura 88 – Resposta consoante o tipo de veículo escolhido (I)

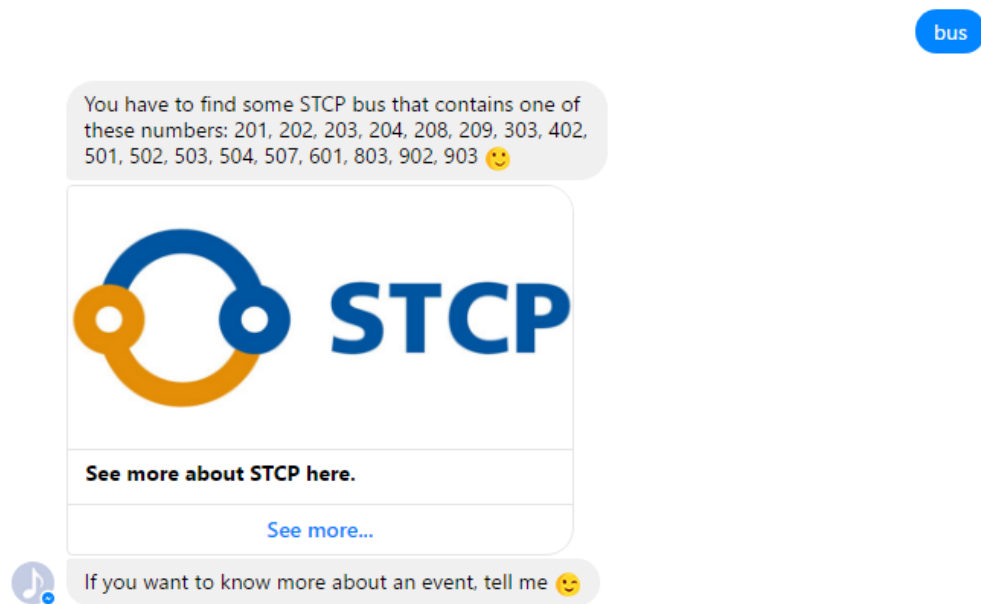


Figura 89 - Resposta consoante o tipo de veículo escolhido (II)

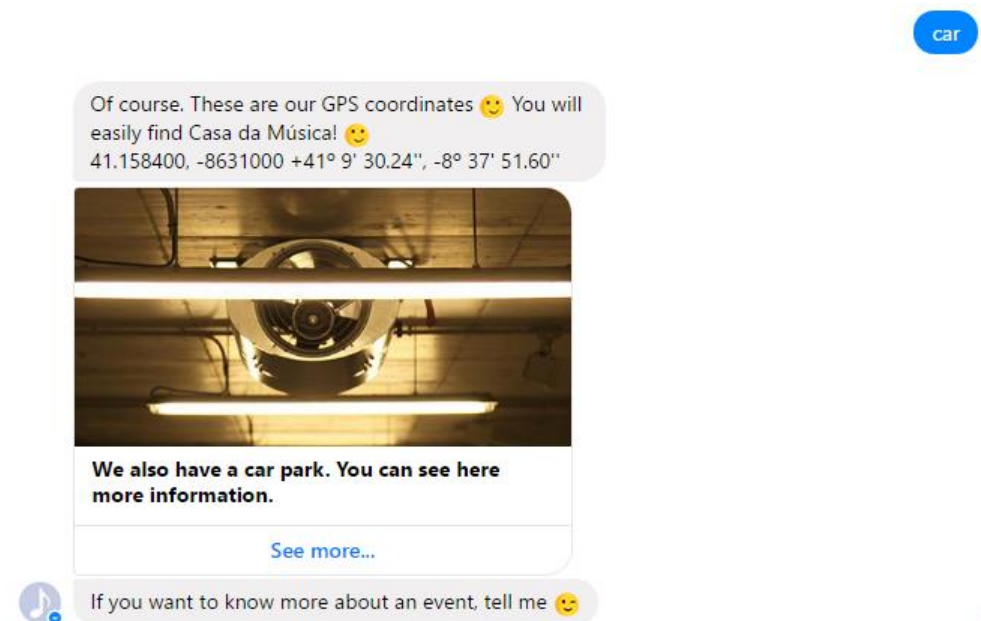


Figura 90 - Resposta consoante o tipo de veículo escolhido (III)

É possível em cada uma das escolhas, visitar um sítio *web* para mais detalhes. No caso do metro, o utilizador pode visitar o sítio *web* do metro do Porto. Caso escolha o autocarro, pode obter mais informações relativas à STCP. E caso pretenda mais detalhes como chegar até à instituição, através do seu próprio carro, são fornecidos mais detalhes sobre o parque da Casa da Música.

4.6.2.3. Horário

A nível do horário, foi criado apenas um *intent* (*getScheduleCM*), onde mais uma vez foram definidas algumas frases que o utilizador poderá dizer, de modo ao agente entender aquilo que poderá ser necessário para despoletar esta ação. E para este pedido, foi definida uma única resposta, que é apresentada na Figura 91.

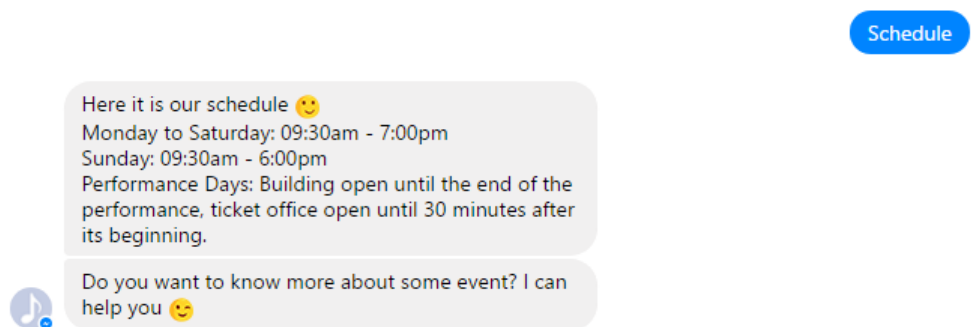


Figura 91 – Conversa relativa ao horário da Casa da Música

4.7. Sugestão de Funcionalidades

De modo a permitir uma conversa entre o agente e o utilizador, e também para dar a conhecer ao utilizador mais funcionalidades desta interface conversacional, no final de alguns pedidos, são enviadas algumas mensagens que refletem outros tipos de pedido que o utilizador pode fazer. Estas frases referem principalmente a funcionalidade de pesquisa de eventos, para levar à compra de bilhetes, que é a funcionalidade principal. Se o pedido for referente à informação sobre eventos, esta resposta extra vai indicar a possibilidade de compra de bilhetes.

Uma conversa entre dois humanos não costuma ficar apenas por uma pergunta e uma resposta, mas tem uma continuidade, ou seja, torna-se mais longa que apenas uma questão e uma afirmação. Também numa interface conversacional deve acontecer isso, para não se tornar numa conversa entre um humano e *robot*.

Vários exemplos foram já apresentados nos subcapítulos referentes às funcionalidades que apresentam esta sugestão de outro tipo de funcionalidades: informações sobre eventos (subcapítulo 4.4), informação sobre artistas (subcapítulo 4.5), informações sobre a Casa da Música (subcapítulo 4.6). Contudo, as figuras seguintes representam algumas dessas respostas que são enviadas ao utilizador (Figura 92).

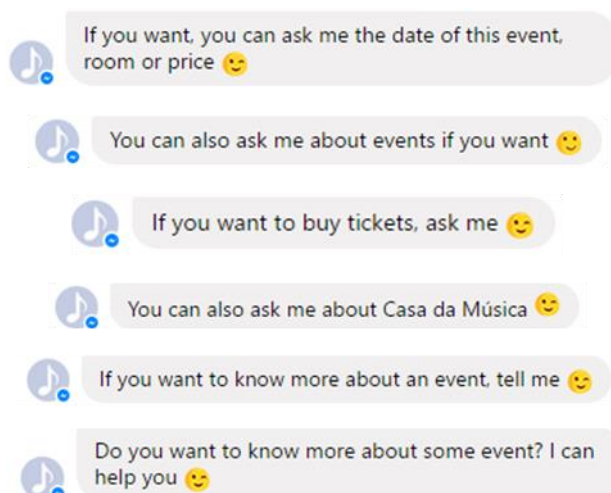


Figura 92 – Sugestão de funcionalidades

Estas sugestões podem levar ao utilizador a conhecer um pouco mais as vantagens do uso deste protótipo e perceber o que este pode fazer.

4.8. Síntese do Capítulo

Neste capítulo foram abordados todos os detalhes de implementação do protótipo funcional, ao longo da fase de desenvolvimento desta dissertação.

Duas componentes, detalhadas primeiramente, foram a componente da ferramenta selecionada para criação do *chatbot* (Api.ai), assim como a componente do *back-end*, que faz a ponte entre a ferramenta e o *web service* da Casa da Música que disponibiliza as informações relativas a artistas e a eventos. Também aqui foi referida como é efetuada a integração com o Facebook Messenger, ou seja, como é que o *chatbot* passa a estar disponível nesta plataforma.

Após indicação de como essas duas componentes foram desenvolvidas numa fase inicial, foram descritas todas as funcionalidades identificadas nos requisitos e casos de uso, a nível de implementação. Aqui estão inseridas todas as entidades que tiveram de ser criadas para desenvolver cada funcionalidade, demonstrações de utilização e outros detalhes pertinentes. E uma vez que foram implementadas sugestões de funcionalidades ao longo da conversa, dadas pelo agente ao utilizador, também este tema foi abordado neste capítulo.

Foram desenvolvidas todas as funcionalidades propostas, utilizando todas as tecnologias também inicialmente definidas.

5. Testes

Neste capítulo são apresentados vários tipos de testes realizados. Os primeiros são os testes operacionais que contribuem para testar a prestação da aplicação. Os segundos são os testes funcionais que permitem perceber se o sistema consegue oferecer todas as funcionalidades esperadas e definidas inicialmente. Por último, são apresentados os testes de usabilidade para testar o nível de aceitação do *chatbot* por parte dos utilizadores, se estes conseguem compreender e manipular facilmente o sistema, e se está a funcionar da forma mais correta para os mesmos.

5.1. Testes Operacionais

Para testar a prestação da aplicação, foram realizados alguns testes relativos ao tempo de resposta do *chatbot*, após a introdução do pedido do utilizador.

Para cada funcionalidade, será apresentado o tempo de resposta relacionado. O tempo de resposta é um valor aproximado, em segundos. As conclusões dos testes realizados para obtenção desse valor estão apresentados na Figura 93 e mais detalhados seguidamente.

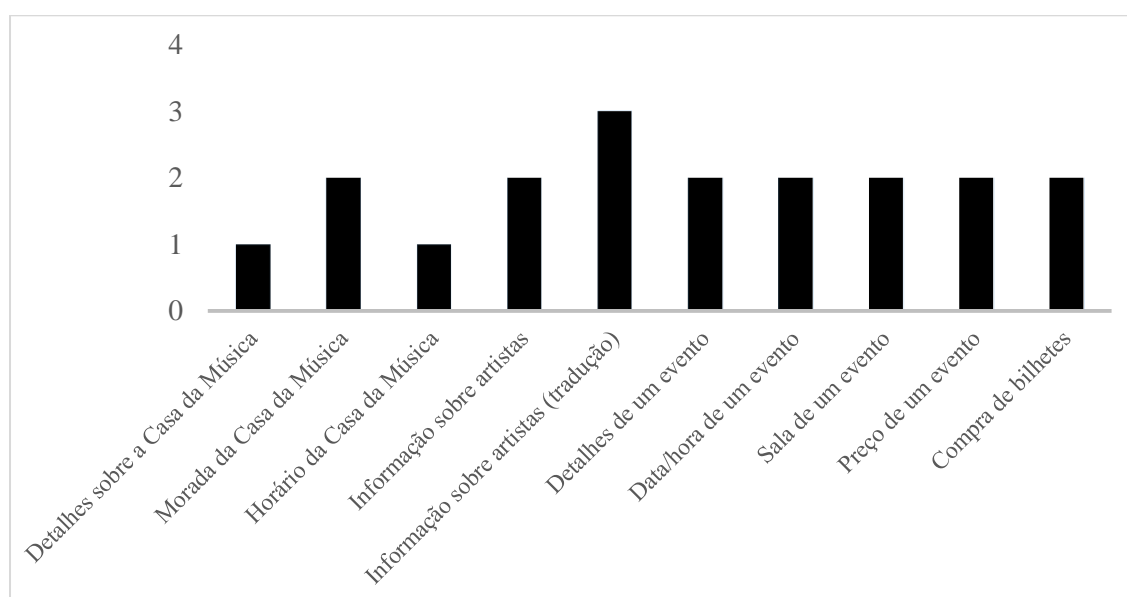


Figura 93 – Tempos de resposta (segundos) associados a cada funcionalidade

Quanto à funcionalidade de obtenção de informações sobre a Casa da Música, a morada é a informação que apresenta um pior tempo de resposta, quando comparada com os outros dois tipos. Esta é a única que não fornece ao utilizador, apenas só mensagens de texto. Para além disso, nenhuma destas funcionalidades recorrem ao *back-end*.

Já a obtenção de informação sobre artistas divide-se em artistas com detalhes em inglês, ou em artistas com possibilidade de adquirir uma tradução automática, pois não têm a informação disponível nesse idioma. O primeiro tipo demora 2 segundos, enquanto que o segundo, mais que o anterior (3 segundos). Mais uma vez este segundo tipo apresenta um cartão com a possibilidade de aceitação de tradução automática, enquanto que o primeiro, apenas texto.

No caso das informações sobre eventos, todos os subtipos apresentam o mesmo tempo de resposta para ser exibida toda a informação pretendida, ou seja, 2 segundos.

Também a compra de bilhetes apresenta um tempo de resposta de 2 segundos, tal como na obtenção de informações sobre eventos.

A conclusão principal que se retira, é que as respostas do lado do agente que são em forma de texto, são mais rápidas que as que apresentam cartões com opções de resposta. Também as funcionalidades que não recorrem ao *back-end*, apresentam um melhor tempo de resposta. A funcionalidade que demora mais tempo, é a obtenção de informação de um artista que tem a possibilidade de tradução automática através do Google Translate.

No geral os tempos de resposta apresentados são aceitáveis, uma vez que num caso real de uma conversa entre duas pessoas, estes tempos seriam bastante maiores, pois teria de ser cotado o tempo de escrita da pessoa que daria a resposta. Assim, o utilizador não necessita esperar muito tempo para que a sua resposta seja dada.

5.2. Testes Funcionais

Os testes funcionais realizados estão diretamente relacionados com todos os casos de uso definidos no subcapítulo 3.2.3, ou seja, todas as funcionalidades do protótipo do *chatbot*, por ordem inversa à sua definição. A primeira funcionalidade a ser abordada é o pedido de informações sobre a Casa da Música, seguida das informações sobre artista, eventos e por fim a compra de bilhetes.

O objetivo destes testes é perceber se o sistema consegue oferecer todas as funcionalidades esperadas.

Começando pelo caso de uso de obtenção de informações sobre a instituição, o primeiro teste relaciona-se com o fornecimento dos detalhes históricos da mesma (Tabela 35).

Tabela 35 – Teste do pedido de informação histórica da Casa da Música

| Informação histórica da Casa da Música | |
|--|---|
| Objetivo | Verificar o correto funcionamento do pedido relativo à informação histórica da Casa da Música. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende saber mais sobre a Casa da Música. |
| Resultado | O utilizador recebe uma resposta que indica factos históricos da Casa da Música. |

Este foi realizado mais do que uma vez, de modo a perceber se diversas respostas eram enviadas ao utilizador, porque assim tinham sido definidas.

Continuando no mesmo caso de uso, o seguinte teste é relativo ao pedido da morada da Casa da Música (Tabela 36). Este relaciona-se ainda com o teste seguinte (Tabela 37).

Tabela 36 – Teste relativo ao pedido da morada da Casa da Música

| Morada da Casa da Música | |
|---------------------------------|---|
| Objetivo | Verificar o correto funcionamento do pedido relativo à morada da Casa da Música. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende saber a morada da Casa da Música. |
| Resultado | O utilizador recebe uma resposta que indica a morada da instituição. Para além disso é feita uma questão ao utilizador, perguntando se este pretende saber como se deslocar até ao local (Tabela 37). |

E como indicado, relacionado com o teste anterior, foi realizado outro teste (Tabela 37) para testar o funcionamento da ajuda do agente para com o utilizador, na medida em que lhe indica como chegar até à instituição, tendo em conta um meio de transporte associado.

Tabela 37 - Teste relativo à ajuda de indicação de como chegar à Casa da Música

| Como chegar à Casa da Música | |
|-------------------------------------|--|
| Objetivo | Verificar o correto funcionamento da aceitação de ajuda de como chegar à Casa da Música. |
| Método de teste | Manual |
| Cenário | 1. Ter iniciado o pedido relativo à morada da Casa da Música. 2. Aceitar a ajuda de como chegar ao local, ou seja, carregar no botão “Sim”. 3. Escolher um meio de transporte (metro, autocarro ou carro). |
| Resultado | O utilizador recebe, conforme o veículo escolhido, como chegar até à Casa da Música. |

Já noutro caso de uso (obter informações sobre artistas), foram também realizados alguns testes. O primeiro está relacionado com um artista com detalhes diretos, ou seja, o utilizador pede informações sobre este e são devolvidas (Tabela 38).

Tabela 38 – Teste à devolução direta de informação de um artista.

| Informação direta de um artista | |
|---------------------------------|---|
| Objetivo | Verificar o correto funcionamento da devolução direta de informação de um artista. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende obter detalhes sobre um artista (que tem descrição). |
| Resultado | O utilizador recebe a descrição referente ao artista em questão. |

Um artista pode também não ser encontrado, porque não se encontra no *web service* da Casa da Música. Para este caso foi realizado um teste (Tabela 39).

Tabela 39 - Teste ao pedido de informação de um artista não existente

| Informação de um artista não existente | |
|--|---|
| Objetivo | Verificar o correto funcionamento da devolução de informação de um artista que não é encontrado. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende obter detalhes sobre um artista (não existente). |
| Resultado | O utilizador recebe uma mensagem indicando que o artista não foi encontrado. |

O seguinte está relacionado com o pedido de informações sobre um artista que não tem detalhes associados, nem em inglês, nem em português (Tabela 38).

Tabela 40 – Teste ao pedido de informação de um artista sem detalhes

| Informação de um artista sem detalhes | |
|---------------------------------------|--|
| Objetivo | Verificar o correto funcionamento quanto ao pedido de informação sobre artistas que não têm detalhes nem em inglês, nem em português. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende obter detalhes sobre um artista (sem descrição em qualquer idioma). |
| Resultado | O utilizador recebe uma mensagem indicando que não tem quaisquer detalhes do artista em questão. |

Também se um artista tiver detalhes disponíveis em português, mas não em inglês, é despoletada uma ação diferente. Assim, foi realizado um teste quanto a esta questão (Tabela 41).

Tabela 41 – Teste ao pedido de informação de um artista sem detalhes em inglês

| Informação de um artista sem detalhes em inglês | |
|--|--|
| Objetivo | Verificar o correto funcionamento quanto ao pedido de informação sobre artistas que não têm detalhes nem em inglês, mas sim em português. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende obter detalhes sobre um artista (sem descrição em inglês). |
| Resultado | O utilizador recebe uma mensagem indicando que não tem informações sobre o artista em inglês, e perguntando se pretende uma tradução automática através do Google Translate, dos detalhes que possui em português. |

Caso o utilizador aceite o pedido de tradução automática, através do Google Translate, da descrição que possui em português, essa é apresentada. Para verificar se ocorria como o esperado, foi também realizado um teste a esta opção (Tabela 42).

Tabela 42 – Teste relativo à tradução dos detalhes de um artista

| Tradução de detalhes de um artista | |
|---|---|
| Objetivo | Verificar o correto funcionamento da aceitação da proposta de tradução, através do Google Translate, dos detalhes disponíveis em português de um artista. |
| Método de teste | Manual |
| Cenário | 1. Ter iniciado o pedido relativo à informação de um artista (sem descrição em inglês). 2. Aceitar a sugestão de tradução automática. |
| Resultado | O utilizador recebe um cartão com uma ligação para o sítio <i>web</i> Google Translate, com o texto a ser traduzido de português para inglês. |

O caso de uso seguinte diz respeito à obtenção de informações sobre eventos. Foram realizados testes quanto ao pedido de detalhes, data/hora, sala e preço de bilhetes. Como todos os tipos de informação que podem ser apresentados, têm um comportamento semelhante, é apresentado na Tabela 43, um teste generalizado.

Tabela 43 - Teste relativo ao pedido da informação de um evento

| Detalhes de um evento | |
|-----------------------|--|
| Objetivo | Verificar o correto funcionamento quanto ao pedido de informação sobre um evento (detalhes, data/hora, sala e preço de bilhetes). |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende obter informação sobre um evento, especificando se quer detalhes, data/hora, sala ou preço dos bilhetes. |
| Resultado | Se o evento não tiver contexto e não for referido, o agente pergunta qual o nome do mesmo. Só depois são apresentados os detalhes pedidos. Caso haja contexto, ou o evento tenha sido referido no pedido, é imediatamente apresentada a informação requerida. |

Se um evento não for encontrado, é enviado outro tipo de resposta. Para tal foi também testado este caso (Tabela 44).

Tabela 44 - Teste relativo ao pedido da informação de um evento não encontrado

| Detalhes de um evento não encontrado | |
|--------------------------------------|--|
| Objetivo | Verificar o correto funcionamento quanto ao pedido de informação sobre um evento não encontrado. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende obter informação sobre um evento (não existente). |
| Resultado | Se o evento não for encontrado, é enviada uma mensagem indicando que o agente não encontrou tal evento. |

Por último, também foram realizados alguns testes para o caso de uso de compra de bilhetes, seguindo o fluxo deste processo. O início deste é apresentado através do primeiro teste (Tabela 45).

Tabela 45 - Teste relativo ao pedido da compra de bilhetes

| Compra de bilhetes | |
|--------------------|---|
| Objetivo | Verificar o correto funcionamento quanto ao pedido de compra de bilhetes para um evento. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que pretende comprar bilhetes para um evento. |
| Resultado | Se o evento não tiver contexto e não for referido, o agente pergunta qual o nome do mesmo. Do mesmo modo, se o utilizador não referir o número de bilhetes que pretende, é feita uma questão, para este indicar qual o número. É apresentado o preço e um cartão para confirmação do utilizador, quanto ao procedimento de compra. |

Após o utilizador ter iniciado o processo de compra de bilhetes, este tem que aceitar continuar este mesmo processo (Tabela 46).

Tabela 46 - Teste relativo à aceitação da compra de bilhetes

| Aceitação da compra de bilhetes | |
|---------------------------------|--|
| Objetivo | Verificar o correto funcionamento quanto à aceitação do procedimento da compra de bilhetes para um evento. |
| Método de teste | Manual |
| Cenário | 1. Ter iniciado o pedido relativo à compra de bilhetes. 2. Aceitar o processo de compra de bilhetes. |
| Resultado | É apresentado novamente um cartão, mas desta vez para escolher a forma de pagamento. |

Depois desta fase, é necessária a escolha da forma de pagamento que o utilizador pretende, ou seja, cartão de crédito/débito ou multibanco (Tabela 47).

Tabela 47 – Teste relativo à escolha de forma de pagamento

| Escolha da forma de pagamento | |
|--------------------------------------|--|
| Objetivo | Verificar o correto funcionamento quanto à escolha da forma de pagamento (cartão de crédito/débito ou multibanco). |
| Método de teste | Manual |
| Cenário | 1. Ter iniciado o processo de aceitação da compra de bilhetes. 2. Escolher a forma de pagamento. |
| Resultado | São apresentadas mensagens que referem os dados que o utilizador deve inserir. |

Para concluir a compra de bilhetes, é necessário inserir todos os dados necessários, consoante a escolha anterior da forma de pagamento. Este foi o último teste funcional a esta funcionalidade (Tabela 48).

Tabela 48 - Teste relativo à inserção dos dados para efetuar compra de bilhetes

| Inserção de dados para efetuar compra de bilhetes | |
|--|--|
| Objetivo | Verificar o correto funcionamento quanto à inserção dos dados requeridos para conclusão da compra de bilhetes. |
| Método de teste | Manual |
| Cenário | 1. Ter iniciado o processo de escolha da forma de pagamento. 2. Indicar todos os dados requeridos. |
| Resultado | No final é apresentada uma mensagem que indica a conclusão do processo de compra de bilhetes. |

Para testar o que acontecia se um utilizador escrevesse algo como “Não quero informações sobre Paulo Barros”, ou seja, fazendo um pedido com uma negação (Tabela 49).

Tabela 49 – Teste relativo a um pedido realizado com negação

| Pedido com negação | |
|--------------------|--|
| Objetivo | Verificar se ao pedir algo, mas com negação, a informação é devolvida na mesma. Neste caso vai ser descrito um caso relativo a informações sobre um artista, mas o resultado é sempre o mesmo independentemente da funcionalidade. |
| Método de teste | Manual |
| Cenário | 1. Abrir uma conversa com o agente, no Facebook Messenger. 2. Escrever uma mensagem, indicando que não pretende obter informações sobre um artista. |
| Resultado | É apresentada na mesma a informação relativa ao artista referido, mesmo que o utilizador tenha dito que não pretende saber. |

Independentemente de o utilizador ter referido que não pretende obter informações sobre algo, a informação é enviada como se o pedido tivesse sido realmente realizado. A ferramenta Api.ai não permite que sejam adicionadas frases que o utilizador possa dizer, e que não seja para enviar a informação. Também os algoritmos de *machine learning* da Api.ai não reconhecem que uma negação foi especificada. Assim sendo, é algo que não é possível controlar.

5.3. Testes de Usabilidade

Para os testes de usabilidade foi aplicado o método *Hallway Testing*. Neste método os participantes são escolhidos aleatoriamente, sem ser definido nenhum grupo ou restrição quanto aos mesmos. Um dos seus objetivos é encontrar falhas que possam existir e que podem ser detetadas por qualquer tipo de utilizador, sem ser necessário um utilizador cuidadosamente escolhido. Enquanto estes estão a testar, normalmente os *designers* desempenham o papel de observadores e tomam notas do que vão observando, tendo em conta a interação do utilizador com o sistema (Desai & Srivastava, 2016).

Para a definição dos participantes, foi colocado num grupo do Facebook do Mestrado de Multimédia, uma mensagem pedindo a disponibilidade de quem pretendesse realizar este teste. Nessa publicação estava presente também uma data, horas e uma sala na FEUP, onde seria efetuado o mesmo. Também foram selecionados alguns amigos.

Para tal, primeiramente foi definida uma lista de tarefas para os utilizadores cumprirem no momento do teste do *chatbot*. A pessoa que observou, neste caso a aluna desta dissertação, seguiu os passos indicados na Tabela 50, para cada teste de usabilidade que realizou.

Tabela 50 – Passos realizados em cada teste de usabilidade

| Passos |
|--|
| 1. Explicação da dissertação, contextualizando. |
| 2. Filmagem do teste. |
| 3. Indicação da tarefa a realizar, pela ordem definida. |
| 4. O utilizador realiza a tarefa indicada. |
| 5. Após término de todas as tarefas, o utilizador responde a um questionário, para medir o seu nível de aceitação. |

Todas as tarefas que foram realizadas aos utilizadores estão apresentadas na Tabela 51, por ordem de realização.

Tabela 51 – Lista de tarefas para os testes de usabilidade

| Tarefas |
|---|
| 1. Pedir informações sobre a Casa da Música. |
| 2. Pedir a morada da instituição e depois aceitar a ajuda de como chegar até ao local. |
| 3. Pedir o horário da Casa da Música. |
| 4. Pedir informações sobre o artista Martin Klett. |
| 5. Pedir informações sobre o artista Paulo Barros e ver a tradução automática. |
| 6. Pedir informações sobre o evento Piano Caos. |
| 7. Pedir a sala. |
| 8. Pedir a data. |
| 9. Pedir o preço dos bilhetes. |
| 10. Comprar 3 bilhetes e depois escolher a forma de pagamento de multibanco. |
| 11. Comprar 6 bilhetes para o evento Fado at the table e depois escolher a opção de pagamento de cartão de crédito. |

Essas tarefas partiram das informações que são possíveis adquirir sobre a instituição, informações sobre artistas com descrição direta e tradução automática, informações sobre eventos e por fim, compra de bilhetes.

Os testes de usabilidade foram realizados a 11 pessoas no total: 7 portugueses, 2 brasileiros, 1 espanhol e 1 britânico, com idades compreendidas entre os 18 e os 35 anos. 55% das pessoas do sexo feminino, e as restantes do masculino, desde estudantes, engenheiros, designers, técnicos de som, produtores audiovisuais, até professores de inglês.

Os testes foram melhorando à medida da sua execução. Nos primeiros foram detetadas algumas falhas a nível de resposta, e foram realizadas melhorias de imediato, após essa constatação. Nem sempre os contextos estavam ativos, porque a Api.ai não associava os pedidos ao *intent* correto. Quantos mais exemplos de afirmações que o utilizador poder recorrer,

associadas a um pedido (*intent*), mais facilmente a Api.ai consegue entender o que o utilizador pretende. Isto acontece porque o agente aprende através dos dados providenciados. Assim sendo, foram acrescentadas outras frases, e do meio para os últimos testes, já não ocorreram falhas. Assim, os testes correram de uma melhor forma, à medida que se foi testando com os utilizadores.

O parecer dos participantes durante os mesmos foi bastante positivo. Alguns utilizadores iam comentando relativamente ao que realmente estavam a gostar, e claro, também os momentos que correram menos bem, principalmente nos primeiros testes. Por isso é que este tipo de testes são úteis, para perceber as necessidades do utilizador e qual a opinião dos mesmos.

Após a realização dos mesmos, cada utilizador respondeu então a um questionário, através do Google Forms, para perceber a aceitação dos mesmos quanto ao protótipo desenvolvido. Neste estavam presentes três questões com uma componente quantitativa com escala (1-5), três de escolha múltipla, duas de resposta aberta e quatro de carácter pessoal.

Iniciando pelo primeiro grupo de questões quantitativas, a Tabela 52 apresenta-as, assim como a escala associada a cada uma.

Tabela 52 – Questões de componente quantitativa - questionário final

| ID | Questão | Escala |
|-----|--|--|
| Q01 | O Facebook Messenger é uma boa plataforma para este <i>chatbot</i> ? | Má Plataforma – Boa Plataforma (1 – 5) |
| Q02 | No geral, gostou deste <i>chatbot</i> ? | Não gostei nada – Gostei muito (1 – 5) |
| Q03 | Se frequentasse a Casa da Música, qual seria o seu nível de aceitação quanto a este <i>chatbot</i> ? | Baixo – Alto (1 – 5) |

Tal como no questionário inicial, aqui também foi utilizado o *software* IBM SPSS Statistics, para retirar as médias, desvios-padrões, e as frequências associadas a cada uma das questões. Todos estes dados estão presentes nos anexos (subcapítulos 8.3 e 8.4).

Quanto à primeira questão, todas as pessoas que participaram nos testes de usabilidade, concordaram que o Facebook Messenger é uma boa plataforma para este *chatbot*. Assim houve unanimidade, com uma percentagem de concordância de 100%, como é possível verificar na Figura 94.

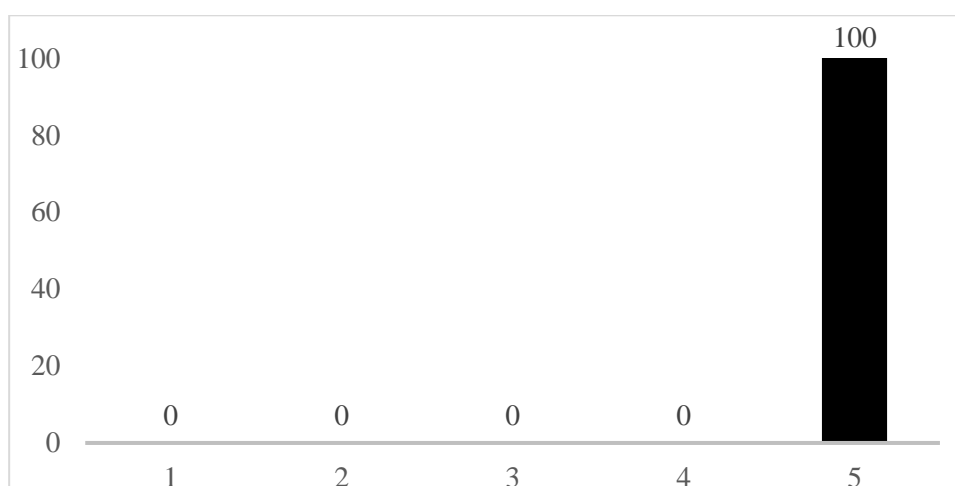


Figura 94 - Percentagem de respostas à questão Q01 - questionário final

Quanto à questão que reflete no geral, o quanto o utilizador gostou do *chatbot*, notou-se alguma variância, mas mesmo assim as respostas não dispersaram muito - Figura 95.

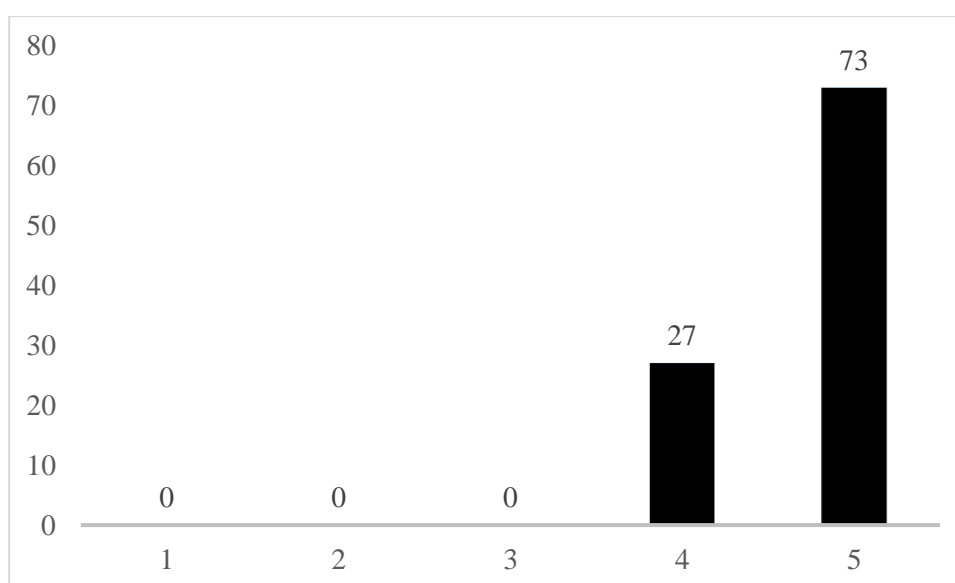


Figura 95 – Percentagem de respostas à questão Q02 - questionário final

A maioria dos participantes gostaram muito deste *chatbot*, sendo que três pessoas, apesar de terem gostado, não responderam com a escala máxima. Contudo, foi um ótimo parecer na sua globalidade.

Uma das perguntas mais importantes, ainda nas questões de componente quantitativa, é a que questiona o nível de aceitação do utilizador quanto ao *chatbot*, se frequentasse a Casa da Música - Figura 96.

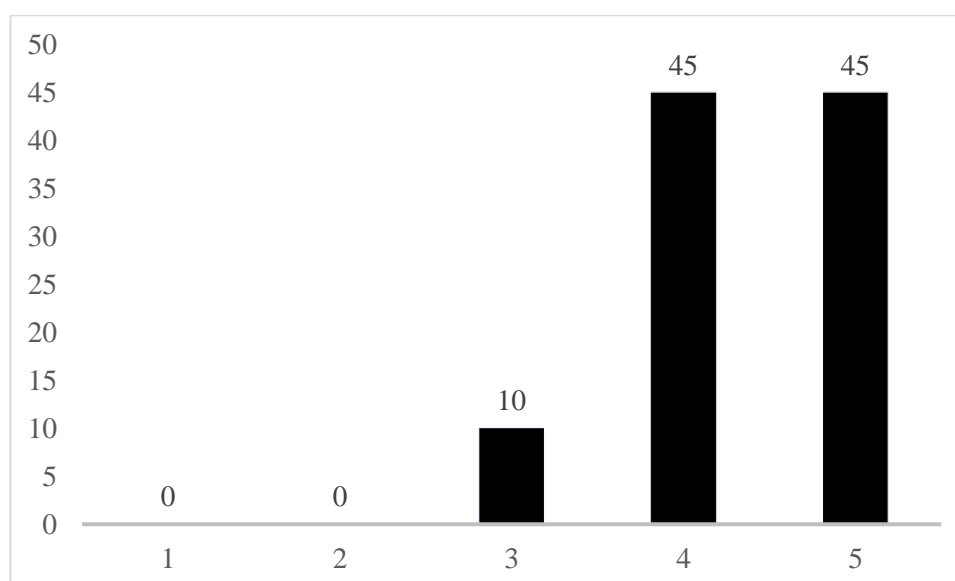


Figura 96 - Percentagem de respostas à questão Q03 - questionário final

Como se pode constatar, o nível de aceitação por parte dos utilizadores foi bastante positivo, sendo que não houve nenhuma resposta que reflita um valor negativo. 90% das pessoas referenciaram um alto nível de aceitação (4 e 5), e só apenas 10%, ou seja, 1 pessoa é que respondeu o valor neutro (3).

Para completar estas informações, foram realizadas outras três questões (Tabela 53), mas de escolha múltipla.

Tabela 53 – Questões de escolha múltipla - questionário final

| ID | Questão | Possíveis Respostas |
|-----|---|---|
| Q04 | Qual a funcionalidade que lhe despertou mais interesse? | Compra de bilhetes Informações sobre eventos Informações sobre artistas Informações sobre a Casa da Música |
| Q05 | Qual a funcionalidade com mais utilidade? | Compra de bilhetes Informações sobre eventos Informações sobre artistas Informações sobre a Casa da Música |
| Q06 | Utilizaria este <i>chatbot</i> se passasse a fazer parte da Casa da Música? | Sim Não Talvez |

Para avaliar o interesse e utilidade relativamente às funcionalidades desenvolvidas, foram efetuadas duas questões (Q04 e Q05 - Figura 97 e Figura 98).

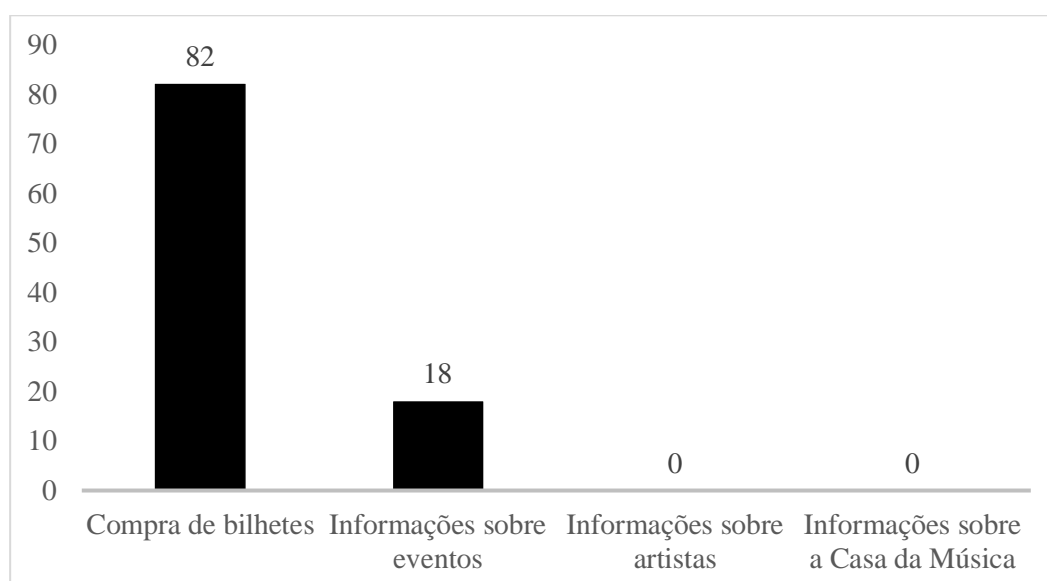


Figura 97 - Percentagem de respostas à questão Q04 - questionário final

A funcionalidade que aos participantes despertou um maior interesse, foi a compra de bilhetes para eventos da Casa da Música (82%), que é a que representa um maior nível de importância neste protótipo. Também alguns participantes (18%), apesar de representarem uma minoria, selecionaram a funcionalidade de informações de eventos, como sendo a mais interessante.

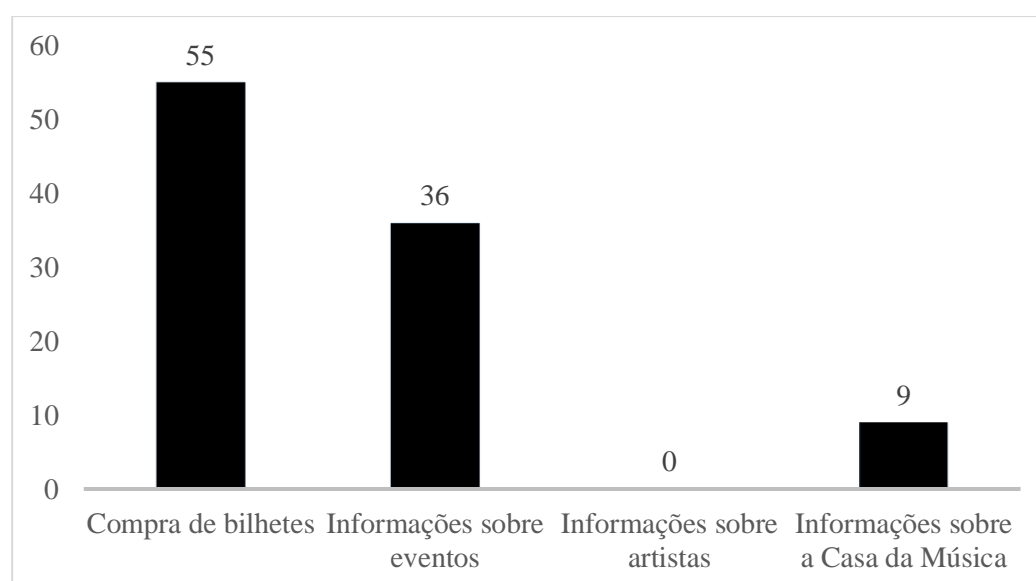


Figura 98 - Percentagem de respostas à questão Q05 - questionário final

Contudo, apesar de o maior interesse coincidir com essas funcionalidades, os participantes apresentaram quanto à utilidade, um cenário um pouco diferente. Nesta já existe uma maior proximidade entre a percentagem refletida quanto às duas primeiras funcionalidades (55% e 36%), e ainda 9% referiu um maior nível de utilidade quanto à possibilidade de obtenção de informações sobre a instituição.

Com a análise destas duas questões, conclui-se que a funcionalidade de informações sobre artistas não apresenta muita relevância, do ponto de vista do utilizador.

Foi também realizada uma questão para avaliar se o utilizador utilizaria o *chatbot*, caso passasse a fazer parte da Casa da Música (Q06 - Figura 99).

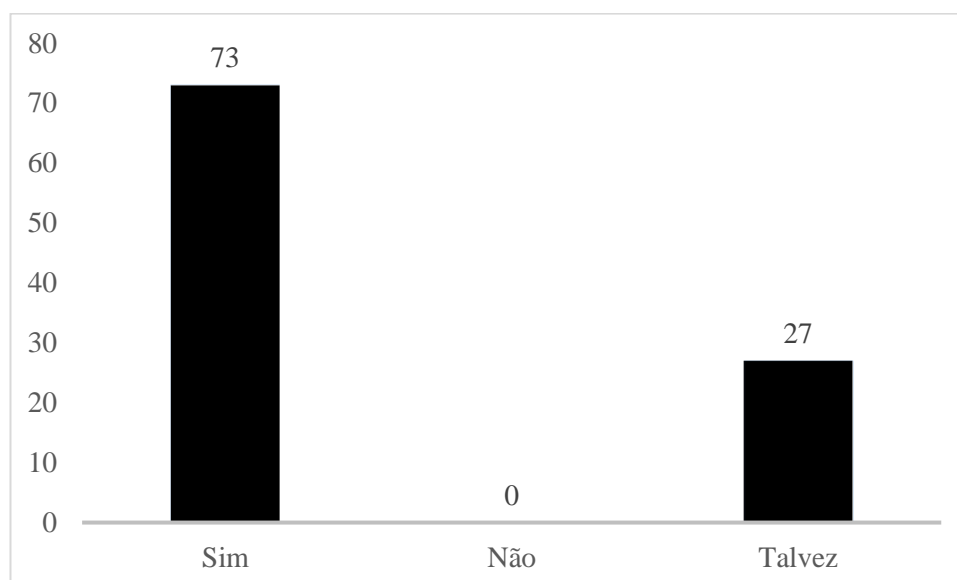


Figura 99 - Percentagem de respostas à questão Q06 - questionário final

Nenhum dos participantes referiu que não usaria este *chatbot*, pois as respostas variaram entre o “sim” e o “talvez”. No primeiro caso, a percentagem apresentada é 73%, enquanto que no segundo, 27%. Isto reflete que uma grande maioria iria utilizar este protótipo se passasse a fazer parte da Casa da Música.

O questionário também era composto por duas questões de resposta aberta. A Tabela 54 lista essas mesmas questões.

Tabela 54 – Questões de resposta aberta - questionário final

| ID | Questão |
|-----|--|
| Q07 | Acha que há algo que podia ser melhorado? |
| Q08 | Gostou de algo em particular que queira referir? |

Algumas das melhorias sugeridas mais direcionadas a falhas, foram corrigidas no decorrer dos testes, sendo que os últimos participantes não se depararam com essas falhas. Contudo, estas melhorias estão também incluídas na Tabela 55, e são apresentadas por ordem de participação. Nem todos os participantes responderam a esta questão, por não ser obrigatória.

Tabela 55 – Melhorias sugeridas pelos participantes

| Melhorias sugeridas |
|---|
| 1. Talvez ter um vídeo no fim da informação que peço sobre um artista ou evento para assim poder ouvir, por exemplo, uma música que possa não conhecer. |
| 2. Talvez aumentar a quantidade de palavras ou expressões que <i>chatbot</i> é capaz de reconhecer, e uma versão em português (poder selecionar o idioma que prefere utilizar). Também alguma informação sobre o sigilo de dados pessoais (como cartão de crédito e outras informações na hora da compra dos bilhetes). |
| 3. Nem sempre os eventos tinham contexto. |
| 4. Reconhecer o nome próprio. |
| 5. Algumas perguntas simples tiveram de ser reformuladas algumas vezes para obter a resposta correta. |
| 6. Não devia pedir os detalhes dos cartões de crédito via <i>web</i> , já que o <i>chat</i> fica registado para o futuro. Integração com o <i>paypal</i> seria a escolha certa. |

Uma das preocupações dos participantes foi relativa ao pagamento, pois alguns não concordam com a inserção dos dados do cartão de crédito/débito, por estes ficarem gravados na conversa. Contudo a integração com outros serviços de pagamento não faz parte do âmbito desta dissertação.

Também foi sugerida a inserção de vídeos sobre a informação requerida pelo utilizador, versões noutros idiomas e reconhecimento do nome próprio. As outras melhorias estão relacionadas com detalhes técnicos, das tais falhas que aconteceram durante os testes, mas que no entanto foram corrigidas.

A última questão de resposta aberta (Q08), foi realizada de modo a perceber o que os participantes mais tinham gostado (Tabela 56).

Tabela 56 - Particularidades que o utilizador gostou

| Particularidades que o utilizador gostou |
|--|
| 1. A simulação da simpatia através do uso de <i>smiles</i> . |
| 2. A interação. E dos botões que aparecem para carregar (sugestões, hipóteses de resposta). |
| 3. Achei muito prático e rápido, tanto para informações, quanto para compra de bilhetes. |
| 4. Possibilita a tradução automática através do Google Translate. |
| 5. Do uso de <i>emojis</i> , torna o processo mais credível e agradável. E da rapidez das respostas! |
| 6. Da capacidade do <i>chatbot</i> de manter o contexto da conversa e fazer perguntas relevantes. |
| 7. A rapidez de resposta, simpatia e acessibilidade |
| 8. Parece um trabalho de qualidade em termos gerais. |
| 9. A noção de contexto, interpretação de frases e os fluxos implementados. |

Vários pontos foram referidos por quem testou o *chatbot*, tais como: a simpatia do agente, que também era transmitida através do uso de *emojis*, a interação e rapidez das respostas, a possibilidade de manter o contexto sem que o utilizador tenha de repetir várias vezes a informação, o uso de botões do Facebook Messenger, assim como a possibilidade de tradução automática através do Google Translate.

No geral, os resultados obtidos foram bastante positivos, assim como o parecer e o interesse que os participantes demonstraram. O que se constatou foi que a funcionalidade de informações sobre artistas não foi muito privilegiada, apesar de que pessoalmente, e também através do questionário, os participantes tenham dito que gostaram bastante por exemplo da tradução automática. A compra de bilhetes foi a funcionalidade com mais utilidade e interesse despertado e o nível de aceitação foi também mais um ponto positivo.

5.4. Síntese do Capítulo

Durante o decorrer desta dissertação, no âmbito do desenvolvimento do sistema de diálogo, foram realizados variados testes para analisar se tudo estava de acordo com o planeado e esperado. Contudo, para mais detalhe, neste capítulo foram descritos testes mais específicos: testes operacionais, funcionais e de usabilidade.

Os testes operacionais permitiram medir os tempos de resposta para cada funcionalidade e assim entender qual a prestação do sistema desenvolvido.

Nos testes funcionais, tentou-se compreender se o sistema consegue oferecer todas as funcionalidades esperadas, e por isso estes testes estão diretamente relacionados com os casos de uso e funcionalidades definidas. Com estes foi possível perceber que se o utilizador fizer um pedido ao agente, indicando explicitamente que não quer receber uma determinada informação, essa informação é enviada na mesma. Este detalhe não é possível resolver neste momento porque a ferramenta não o permite, nem os algoritmos de *machine learning* da Api.ai são capazes de entender essa negação. Contudo, os restantes testes foram concluídos com sucesso.

Nos testes de usabilidade participaram 11 utilizadores de diferentes nacionalidades, de modo a testarem o *chatbot* e perceber qual o nível de aceitação dos mesmos, quanto ao protótipo. Estes testes tinham tarefas definidas para o utilizador não se dispersar, mas dando-lhe liberdade quanto à forma de como os pedidos eram realizados. Assim, os testes permitiram fazer melhorias e retirar conclusões bastante úteis.

Segundo os testes operacionais, os tempos de resposta apresentavam valores relativamente baixos, o que significa que os utilizadores não necessitavam de muito tempo de espera pelas respostas. Isto ficou provado através dos testes de usabilidade, nos quais os participantes referiram que ficaram surpreendidos com a rapidez.

6. Conclusões

Neste capítulo serão apresentados os resultados do trabalho desenvolvido, assim como as conclusões retiradas ao longo do mesmo. Começará pela descrição dos objetivos realizados, incluindo o seu grau de concretização e passará para a apresentação do possível trabalho futuro. O último capítulo tem uma componente mais pessoal, pois refere uma apreciação final de toda a dissertação.

6.1. Objetivos Realizados

Um dos principais objetivos desta dissertação consistia numa pesquisa sobre o tema de interfaces conversacionais, de modo a obter um bom nível de conhecimento sobre o tema, as suas vantagens, utilidade, o motivo de utilização das mesmas, e também como evoluíram desde o seu aparecimento. E segundo a área de aplicação (agenda cultural da Casa da Música), o objetivo final era medir o nível de aceitação dos utilizadores quanto ao uso de um protótipo de um *chatbot*, utilizando o Facebook Messenger. Para esta medição, foi necessário também a investigação de plataformas e tecnologias que permitissem criar este protótipo, selecionando uma delas.

Assim sendo, consoante os objetivos do trabalho em questão, será apresentada uma descrição dos mesmos, assim como os seus graus de concretização (Tabela 57).

Tabela 57 – Objetivos e sua concretização

| Objetivo | Grau de concretização (%) |
|---|---------------------------|
| 1. Efetuar uma caracterização detalhada de interfaces conversacionais, em termos tecnológicos e funcionais. | 100 |
| 2. Identificar como se processou a evolução das interfaces conversacionais? | 100 |
| 3. Identificar como é possível as interfaces conversacionais terem utilidade para as pessoas no contexto das suas interações com produtos e serviços. | 100 |
| 4. Identificar quais as ferramentas que melhor permitem a criação de um sistema conversacional, selecionando uma. | 100 |
| 5. Implementar um protótipo tendo em conta os casos de uso definidos: compra de bilhetes, obtenção de informações sobre eventos, artistas e sobre a Casa da Música. | 100 |
| 6. Fazer testes com utilizadores para entender qual o nível de aceitação dos mesmos, relativamente a um sistema conversacional baseado numa área de aplicação específica (agenda cultural da Casa da Música). | 100 |

Todos os objetivos definidos inicialmente foram concluídos com sucesso e foram abordados ao longo deste documento. Os primeiros quatro objetivos foram explorados no capítulo do Estado da Arte. O quinto objetivo foi descrito nos capítulos de Implementação, Testes Operacionais e Testes Funcionais. Já o último objetivo foi abordado no capítulo de Testes de Usabilidade.

6.2. Trabalho Futuro

Futuramente, caso este *chatbot* fosse posto em prática, há algumas funcionalidades que poderiam ser bastante úteis. Neste momento o utilizador não tem a possibilidade de pedir ao agente eventos para crianças, relacionados com um tema, ou até mesmo num determinado dia ou intervalo de tempo. Do mesmo modo, o utilizador também não consegue pedir artistas relacionados com uma área, mas sim específicos. Estas são algumas funcionalidades que poderiam vir a ser implementadas no futuro, de modo a que o utilizador tenha mais liberdade de escolha, sem ter que saber todos os nomes de artistas e eventos que pretende adquirir mais informações.

A compra de bilhetes também poderia vir a ser concluída, ou seja, permitir mesmo o pagamento da compra de bilhetes. Este processo poderia ser feito, como sugerido nos testes de

usabilidade, através da ligação com *paypal*, pois assim os dados do cartão não ficariam guardados e disponíveis na conversa com o agente.

Como foi referido, o *chatbot* desenvolvido é um protótipo que tem como principal objetivo testar o nível de aceitação dos utilizadores, quanto a esta área de trabalho. Para tal, foram definidas apenas algumas funcionalidades para que os utilizadores conseguissem experimentar o protótipo, assim como permitir testar a ferramenta de criação de *chatbots* selecionada.

6.3. Apreciação Final

O desenvolvimento desta dissertação foi uma mais-valia para conhecimento de um tema que é cada vez mais abordado e bastante atual, assim como para perceção de como os utilizadores reagem a esta atualidade. Há pessoas que não aceitam a substituição de pessoas por *robots* que acabam por imitar os humanos no contexto da interação e comunicação, mas também muitas delas apresentam um nível de aceitação bastante elevado, por ser algo com tanta utilidade. Por exemplo, para além de não haver necessidade de as pessoas acabarem por realizar tarefas rotineiras de resposta aos utilizadores (no caso de um *chat*), também é muito vantajoso para os utilizadores porque não têm de procurar as informações que pretendem (no caso de um sítio *web*). É também normal as pessoas mais jovens apresentarem um nível de aceitação mais elevado que as restantes, pois já estão mais contextualizadas no mundo das tecnologias.

Relativamente à fase de desenvolvimento, foi bastante motivante a sua concretização, pois para além do projeto estar envolvido num tema tão atual, teve também associada uma área de aplicação na qual não existia nada semelhante. O conhecimento adquirido através da pesquisa e do uso de ferramentas anteriormente desconhecidas foi também muito desafiante.

As organizações associadas foram partes muito importantes no desenrolar desta dissertação, de modo a fornecerem todas as ferramentas necessárias. A empresa Xarevision S.A. permitiu uma boa definição do trabalho a desenvolver, assim como um bom seguimento durante todo o percurso, através das constantes reuniões de ponto de situação. A Casa da Música forneceu dados bastante importantes para análise do público-alvo, acesso aos dados internos relativos a artistas e eventos, assim como todo o tipo de ajuda necessária, tanto pessoalmente, como através de mensagens de *email*. Todos os elementos que estiveram de alguma forma relacionados com a dissertação, permitiram uma boa direção e continuidade no projeto.

Através de todas as fases executadas, o trabalho foi realizado com sucesso. Todas os objetivos definidos inicialmente foram cumpridos.

7. Referências

- Agus. (setembro de 2016). *10 Tools to Build your Own Chatbots*. Obtido em 7 de novembro de 2016, de <http://www.hongkiat.com/blog/tools-to-build-chatbots/>
- Amazon Developer. (2014). *Echo & Echo Dot*. Obtido em 28 de dezembro de 2016, de <https://developer.amazon.com/echo>
- An, M. (23 de Maio de 2016). *What is a Bot? How Conversational UI is Changing Tech*. Obtido em 14 de novembro de 2016, de <https://research.hubspot.com/reports/what-is-a-bot#LINK4>
- Api.ai. (2016). *Api.ai*. Obtido em 3 de novembro de 2016, de <https://api.ai/>
- Api.ai. (2016). *Entities*. Obtido em 29 de março de 2017, de <https://docs.api.ai/docs/concept-entities>
- Apple. (2010). *Siri*. Obtido em 28 de dezembro de 2016, de <http://www.apple.com/ios/siri/>
- Blom, A., & Thorsen, S. (2013). *A sentiment-based chat bot*.
- botlist. (2016). *Hi Poncho*. Obtido em 27 de dezembro de 2016, de <https://botlist.co/bots/605-hi-poncho>
- Brownlee, J. (4 de abril de 2016). *Conversational Interfaces, Explained*. Obtido de <https://www.fastcodesign.com/3058546/conversational-interfaces-explained>
- Casa da Música. (2016). *Casa da Música*. Obtido em 12 de dezembro de 2016, de <http://www.casadamusica.com/>
- Catanzariti, P. (2015 de dezembro de 2015). *Five Simple Ways to Build Artificial Intelligence in 2016*. Obtido em 27 de outubro de 2016, de <https://www.sitepoint.com/five-simple-ways-to-build-artificial-intelligence-in-2016/>
- Chatfuel. (2015). Obtido em 3 de novembro de 2016, de <https://chatfuel.com/>
- Desai, S., & Srivastava, A. (2016). *Software Testing: A Practical Approach*. PHI Learning Pvt. Ltd.
- Google. (2016). *Google Allo*. Obtido em 28 de dezembro de 2016, de <https://allo.google.com/>
- Google. (2016). *Google Home*. Obtido em 28 de dezembro de 2016, de <https://madeby.google.com/home/>

- IBM. (2011). *Watson*. Obtido em 28 de dezembro de 2016, de <https://www.ibm.com/watson/>
- KLM. (2016). *Enjoy KLM on Messenger*. Obtido em 27 de dezembro de 2016, de <https://messenger.klm.com>
- Knight, W. (2016). AI Unspoken Problem. *MIT Technology Review*, 119.
- Lun, E. v. (2009). *A.L.I.C.E.* Obtido em 4 de outubro de 2016, de <https://www.chatbots.org/chatbot/a.l.i.c.e/>
- Messenger Chatbots. (2016). *1-800-Flowers Chat Bot Walkthrough Review*. Obtido em 27 de dezembro de 2016, de <http://www.messengerchatbots.com/Reviews/1-800-flowers-com-chat-bot-walkthrough-review.aspx>
- Microsoft. (2014). *Cortana*. Obtido em 28 de dezembro de 2016, de <https://www.microsoft.com/en-us/mobile/experiences/cortana/>
- Microsoft. (2017). *Contact us*. Obtido em 12 de dezembro de 2016, de <https://support.microsoft.com/en-us/contactus/>
- Pinho, C. (2016). *Interação Homem-Computador através de Interfaces Conversacionais*. Obtido em 26 de setembro de 2016
- Rozmovits, B. A. (1996). The Design of User Interfaces for Digital Speech Recognition Software. *Digital Technical Journal*, Vol.8, 117–128.
- Rui Campos, S. (junho de 2017). *Chatbots as a platform*. Obtido em 5 de junho de 2017, de <https://docs.google.com/presentation/d/1KmnkDLatJLzNucufuoOw5nYTf6BIEYtQDm4zfnhVVw/edit>
- Wallace, M. (1999-2006). *Eliza, computer therapist*. Obtido em 4 de outubro de 2016, de <http://www.manifestation.com/neurotoys/eliza.php3>
- Williams, O. (7 de dezembro de 2016). *How voice assistants seemingly came from nowhere*. Obtido em 27 de dezembro de 2016, de <https://medium.com/conversational-interfaces/how-voice-assistants-seemingly-came-from-nowhere-33747876b91f#.upynbry6j>
- Wit.ai. (2016). *Bot Engine*. Obtido em 3 de setembro de 2016, de <https://wit.ai/blog/2016/04/12/bot-engine>
- Wong, J. C. (6 de abril de 2016). *What is a chat bot, and should I be using one?* Obtido em 30 de dezembro de 2016, de <https://www.theguardian.com/technology/2016/apr/06/what-is-chat-bot-kik-bot-shop-messaging-platform>
- Xarevision, S.A. (2015). *Xarevision*. Obtido em 12 de dezembro de 2016, de <http://www.xarevision.pt/>

8. Anexos

8.1. Anexo 1 – Tabelas de Frequências (Questionário Inicial)

| Q01. Qual o seu nível de interesse em eventos na Casa da Música? | | | | | |
|--|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 37 | 12,7 | 12,7 | 12,7 |
| | 2,0 | 48 | 16,4 | 16,4 | 29,1 |
| | 3,0 | 89 | 30,5 | 30,5 | 59,6 |
| | 4,0 | 78 | 26,7 | 26,7 | 86,3 |
| | 5,0 | 40 | 13,7 | 13,7 | 100,0 |
| | Total | 292 | 100,0 | 100,0 | |

| Q02. Qual o seu nível de conhecimento relativo a <i>chatbots</i> ? | | | | | |
|--|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 104 | 35,6 | 35,6 | 35,6 |
| | 2,0 | 54 | 18,5 | 18,5 | 54,1 |
| | 3,0 | 61 | 20,9 | 20,9 | 75,0 |
| | 4,0 | 53 | 18,2 | 18,2 | 93,2 |
| | 5,0 | 20 | 6,8 | 6,8 | 100,0 |
| | Total | 292 | 100,0 | 100,0 | |

| Q03. Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse comprar bilhetes para eventos? | | | | | |
|--|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 13 | 4,5 | 4,5 | 4,5 |
| | 2,0 | 17 | 5,8 | 5,8 | 10,3 |
| | 3,0 | 59 | 20,2 | 20,2 | 30,5 |
| | 4,0 | 108 | 37,0 | 37,0 | 67,5 |
| | 5,0 | 95 | 32,5 | 32,5 | 100,0 |
| | Total | 292 | 100,0 | 100,0 | |

| Q04. Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse obter informações sobre eventos? | | | | | |
|--|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 15 | 5,1 | 5,1 | 12,7 |
| | 2,0 | 22 | 7,5 | 7,5 | 39,7 |
| | 3,0 | 79 | 27,1 | 27,1 | 82,5 |
| | 4,0 | 125 | 42,8 | 42,8 | 100,0 |
| | 5,0 | 51 | 17,5 | 17,5 | 12,7 |
| | Total | 292 | 100,0 | 100,0 | |

| Q05. Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse obter informações sobre artistas? | | | | | |
|---|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 15 | 5,1 | 5,1 | 5,1 |
| | 2,0 | 34 | 11,6 | 11,6 | 16,8 |
| | 3,0 | 90 | 30,8 | 30,8 | 47,6 |
| | 4,0 | 109 | 37,3 | 37,3 | 84,9 |
| | 5,0 | 44 | 15,1 | 15,1 | 100,0 |
| | Total | 292 | 100,0 | 100,0 | |

| Q06. Qual o nível de utilidade que teria este <i>chatbot</i> caso permitisse obter informações sobre a instituição (Casa da Música)? | | | | | |
|--|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 14 | 4,8 | 4,8 | 4,8 |
| | 2,0 | 49 | 16,8 | 16,8 | 21,6 |
| | 3,0 | 103 | 35,3 | 35,3 | 56,8 |
| | 4,0 | 89 | 30,5 | 30,5 | 87,3 |
| | 5,0 | 37 | 12,7 | 12,7 | 100,0 |
| | Total | 292 | 100,0 | 100,0 | |

8.2. Anexo 2 – Domínios Suportados por cada Idioma

| Domínios | Inglês | Francês | Italiano | Japonês | Português | Russo | Espanhol |
|----------------------------|--------|---------|----------|---------|-----------|-------|----------|
| Aplicações | x | | x | | | | |
| Autenticação | x | | x | x | | | |
| Livros | x | | | | | | |
| Calculadora | x | | | | | | |
| Calendário | x | | | | | | |
| Chamadas | x | | x | x | | | |
| Controlo de dispositivo | x | | x | | | | |
| Email | x | | | | | | |
| Eventos | x | | | | | | |
| Finanças | x | | | | | | |
| Horários de voo | x | | | | | | |
| Seleção de idioma | x | | x | | | | |
| Aprendizagem | x | | x | | | | |
| Manuseamento de aplicações | x | | x | | | | |
| Mapas | x | | x | x | x | x | x |
| Media | x | | x | x | x | x | x |
| Mensagens | x | | x | x | | | |
| Notícias | x | | | | | | |
| Notas | x | | | | | | |
| Pontos de interesse | x | | x | x | x | x | x |
| Lembretes e notificações | x | | x | | | | |
| Compras | x | | | | | | |
| Conversa curta | x | x | x | | | | |
| Casa inteligente | x | | | | | | |
| Redes sociais | x | | x | | | | |

| Domínios | Inglês | Francês | Italiano | Japonês | Português | Russo | Espanhol |
|--|--------|---------|----------|---------|-----------|-------|----------|
| Desporto | x | | | | | | |
| Tarefas | x | | | | | | |
| Pesquisa relativa a táxis | x | | | | | | |
| Datas e tempos | x | | x | | | | |
| Tradução | x | x | x | x | x | x | x |
| Listas de televisão | x | | | | | | |
| Unidades e formatos | x | | | | | | |
| Nome de utilizador e nome do agente | x | | x | | | | |
| Meteorologia | x | x | x | x | x | x | x |
| Pesquisa <i>web</i> | x | | x | | | | |

8.3. Anexo 3 – Estatísticas Descritivas (Questionário Final)

| Questão | N | Mínimo | Máximo | Média | Desvio Padrão |
|--|----|--------|--------|-------|---------------|
| O Facebook Messenger é uma boa plataforma para este <i>chatbot</i> ? | 11 | 5,0 | 5,0 | 5,0 | 0,0 |
| No geral, gostou deste <i>chatbot</i> ? | 11 | 4,0 | 5,0 | 4,7 | 0,5 |
| Se frequentasse a Casa da Música, qual seria o seu nível de aceitação quanto a este <i>chatbot</i> ? | 11 | 3,0 | 5,0 | 4,4 | 0,7 |

8.4. Anexo 4 – Tabelas de Frequências (Questionário Final)

| Q01. O Facebook Messenger é uma boa plataforma para este <i>chatbot</i> ? | | | | | |
|---|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 2,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 3,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 4,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 5,0 | 11 | 100,0 | 100,0 | 100,0 |
| | Total | 11 | 100,0 | 100,0 | |

| Q02. No geral, gostou deste <i>chatbot</i> ? | | | | | |
|--|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 2,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 3,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 4,0 | 3 | 27,3 | 27,3 | 27,3 |
| | 5,0 | 8 | 72,7 | 72,7 | 100,0 |
| | Total | 11 | 100,0 | 100,0 | |

| Q03. Se frequentasse a Casa da Música, qual seria o seu nível de aceitação quanto a este <i>chatbot</i> ? | | | | | |
|---|-------|------------|-------------|--------------------|-------------------------|
| | | Frequência | Percentagem | Percentagem válida | Percentagem acumulativa |
| Válido | 1,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 2,0 | 0 | 0,0 | 0,0 | 0,0 |
| | 3,0 | 1 | 9,1 | 9,1 | 9,1 |
| | 4,0 | 5 | 45,5 | 45,5 | 54,5 |
| | 5,0 | 5 | 45,5 | 45,5 | 100,0 |
| | Total | 11 | 100,0 | 100,0 | |